MAKE | BUILD | HACK | CREATE

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.CC | November 2023 | Issue #72

Raspberry Pi 5 PROJECTS



Put the new computer to use

Nov. 2023
Issue #72 £6

BAR CODES
The easiest form of image recognition

ENERGY MONITOR
Save money on your heating bill with a Raspberry Pi Pico

HAXOPHONE
Building an electronic wind instrument

3D PRINTING **ARDUINO** NEURAL NETWORKS

Free eBook!



Download your copy from

hsmag.cc/freecadbook



Welcome to HackSpace magazine

Raspberry Pi 5 is here! Last month, we chatted to its creators about what to expect, and this month, we've been putting it through its paces in a few different projects to find out where it really excels.

The Raspberry Pi team has done a great job of increasing performance across the board. That means that most projects will get a speed boost. If your project is running perfectly fine on its existing hardware, there's obviously no need to upgrade, but if you're itching to add more things and want them running faster (and let's face it, we always want more things running faster), there's a good chance Raspberry Pi 5 will have what you want.

BEN EVERARD
Editor ben.everard@raspberrypi.com

Got a comment, question, or thought about HackSpace magazine?

get in touch at **hsmag.cc/hello**

GET IN TOUCH

hackspace@raspberrypi.com
hackspacemag
hackspacemag

ONLINE

hsmag.cc

Available on the App Store

GET IT ON Google Play



PAGE 34
FREE PICO W
WHEN YOU SUBSCRIBE

EDITORIAL

Editor
Ben Everard
ben.everard@raspberrypi.com

Features Editor
Andrew Gregory
andrew.gregory@raspberrypi.com

Sub-Editors
David Higgs, Nicola King

DESIGN

Critical Media and Raspberry Pi
criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Sam Ribbits, Sara Parodi, Jack Willis

Photography
Brian O'Halloran

CONTRIBUTORS

Marc de Vinck, Andrew Lewis, Thomas Burns, Rob Miles

PUBLISHING

Publishing Director
Brian Jepson
brian.jepson@raspberrypi.com

Advertising
Charlie Milligan
charlotte.milligan@raspberrypi.com

DISTRIBUTION

Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS

Unit 6, The Enterprise Centre,
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE

To subscribe
01293 312189
hsmag.cc/subscribe

Subscription queries
hackspace@subscriptionhelpline.co.uk

This magazine is printed on paper sourced from sustainable forests. The printer operates an environmental management system which has been assessed as conforming to ISO 14001.

HackSpace magazine is published by Raspberry Pi Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISSN: 2515-5148.

Contents



06

06 SPARK

06 **Top Projects**
Big or small, you build them all

18 **Objet 3d'art**
A 3D-printed turntable for your tunes

20 **Letters**
Our wireless headphones have disturbed some

23 LENS

24 **Raspberry Pi Projects**
Put the new computer to use

36 **How I Made:** Alexatron
A beautiful mecanum-wheeled robotics project

44 **Interview:** Jeff Geerling
Progress, open source, automation, and more

Cover Feature



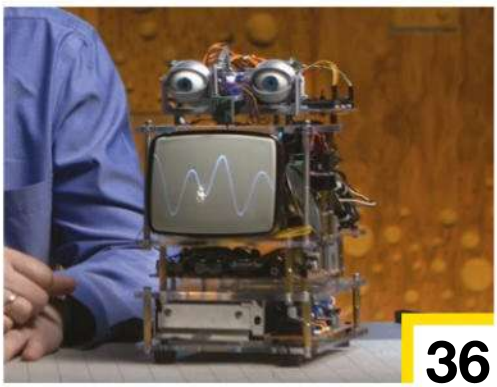
24

Tutorial

Laser cutting



64 Teach a K40 laser cutter to speak proper G-code



36

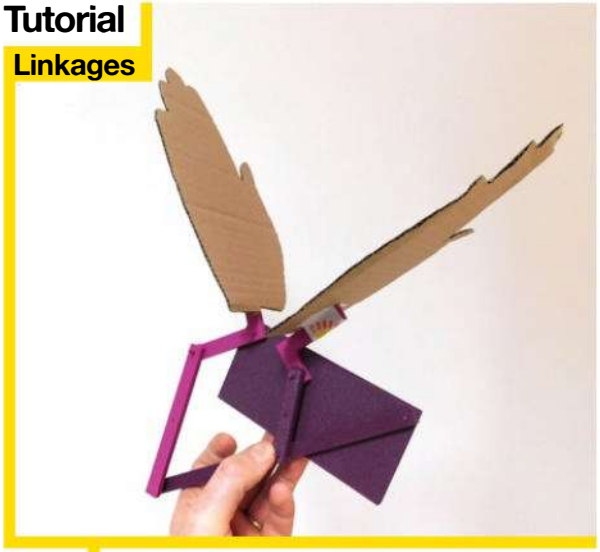


74



82

Tutorial
Linkages



70

Design and 3D-print mechanisms to turn motion into different motion

Crowdfunding
InvenTABLE



96

CAD (Cardboard Aided Design) just got even more professional

53 FORGE

54 **SoM** Synthio
Make music on a Raspberry Pi Pico

58 **Tutorial** Pico
Build a simple true-or-false game with a Pico

64 **Tutorial** Laser cutting
Upgrade your K40 with LightBurn

70 **Tutorial** 3D printing
Design and 3D-print mechanical linkages

74 **Tutorial** Active power monitor
Keep an eye on how much your heating costs

81 FIELD TEST

82 **Best of Breed**
Extract data from black and white lines

88 **Review** Prusa input shaper
The Prusa MK4 gets an upgrade

90 **Review** LibrePCB
Design software for those who love to make PCBs

94 **Review** Kaluma
JavaScript for RP2040 devices

96 **Crowdfunding** InvenTABLE
A sharper, cleaner way to cut cardboard

Some of the tools and techniques shown in HackSpace Magazine are dangerous unless used with skill, experience and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. HackSpace Magazine is intended for an adult audience and some projects may be dangerous for children. Raspberry Pi Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in HackSpace Magazine. Laws and regulations covering many of the topics in HackSpace Magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in HackSpace Magazine may go beyond. It is your responsibility to understand the manufacturer's limits. HackSpace magazine is published monthly by Raspberry Pi Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS, United Kingdom. Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701, is the mailing agent for copies distributed in the US and Canada. Application to mail at Periodicals prices is pending at Williamsport, PA. Postmaster please send address changes to HackSpace magazine c/o Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701.

REGULAR

Diablo robot

By DirectDriveTech

hsmag.cc/Diablo

Here we have another one of the many robotics projects to use ROS, the Robot Operating System, running on a Raspberry Pi 4, and programmed in Python. What makes this one special is that it balances on two wheeled legs, can move at 2 m/s, can carry up to 80kg, and can balance all by itself. It can jump down a couple of stairs and land on its feet like a cat – it's a little unnerving, but it points the way to a future where delivery robots won't be scuppered by having to go up stairs.





Above This makes our attempts at robotics seem rather rudimentary





Zvychai One

By **mkdxdx** hsmag.cc/Zvychai-One

Inspired, in part, by Zack Freedman and his lovely Pi 400 cyber deck, and, in part, by the potential need for a post-doomsday communication device, maker mkdxdx has made this, the Zvychai One. It's a cyber deck, comprising an 8GB Raspberry Pi 4, a 5-inch Eyoyo TFT display, a backlit keyboard, several LEDs, and a beautiful 3D-printed clamshell case. It was the winner of the Hackaday cyber deck competition, and will be just the thing in the event of apocalypse.





Right
Although this device is called the Zvychai One, it's actually the fourth iteration


Eyoyo

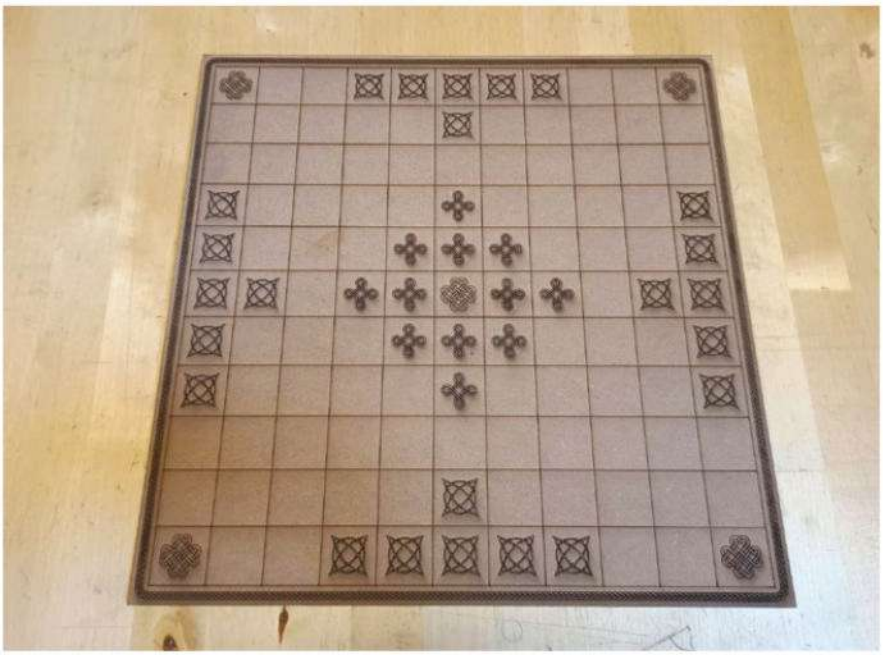
Hnefatafl set

By **Leon Reboul** hsmag.cc/Hnefatafl



Tafl, from the Anglo-Saxon word for table, is a term that encompasses a variety of board games that survive from the early medieval period of northern Europe. There's the Ard Rí variant, from what's now Scotland, Brandudh from Ireland, and, among others, Hnefatafl, from Scandinavia. It's a bit like chess, except that the two players have different objectives: one player uses his forces to try to capture the king, located at the centre of the board, while the other player's objective is to escape, by getting the king to one of the corner squares. It's chess, for Vikings.

This laser-cut Hnefatafl set, by Leon Reboul, uses laser-cut 3 mm plywood rather than the carved bones of Lindisfarne's monks, but it does capture the Viking aesthetic with some gorgeous artwork by Jonas Lau Markussen **(jonaslaumarkussen.com)**. Now, if only we knew how to play…



Right
The laser cutting for this project was done mostly at Fablab Skanderborg, Denmark



Soft robot

By Yoav Matia et al

hsmag.cc/Soft-robot

Hydraulic actuators work the way they do because the fluid under pressure performs uniformly: move a piston 50% of the way along its length, and the connected arm, leg, or whatever, will also move 50% of the way it's built to.

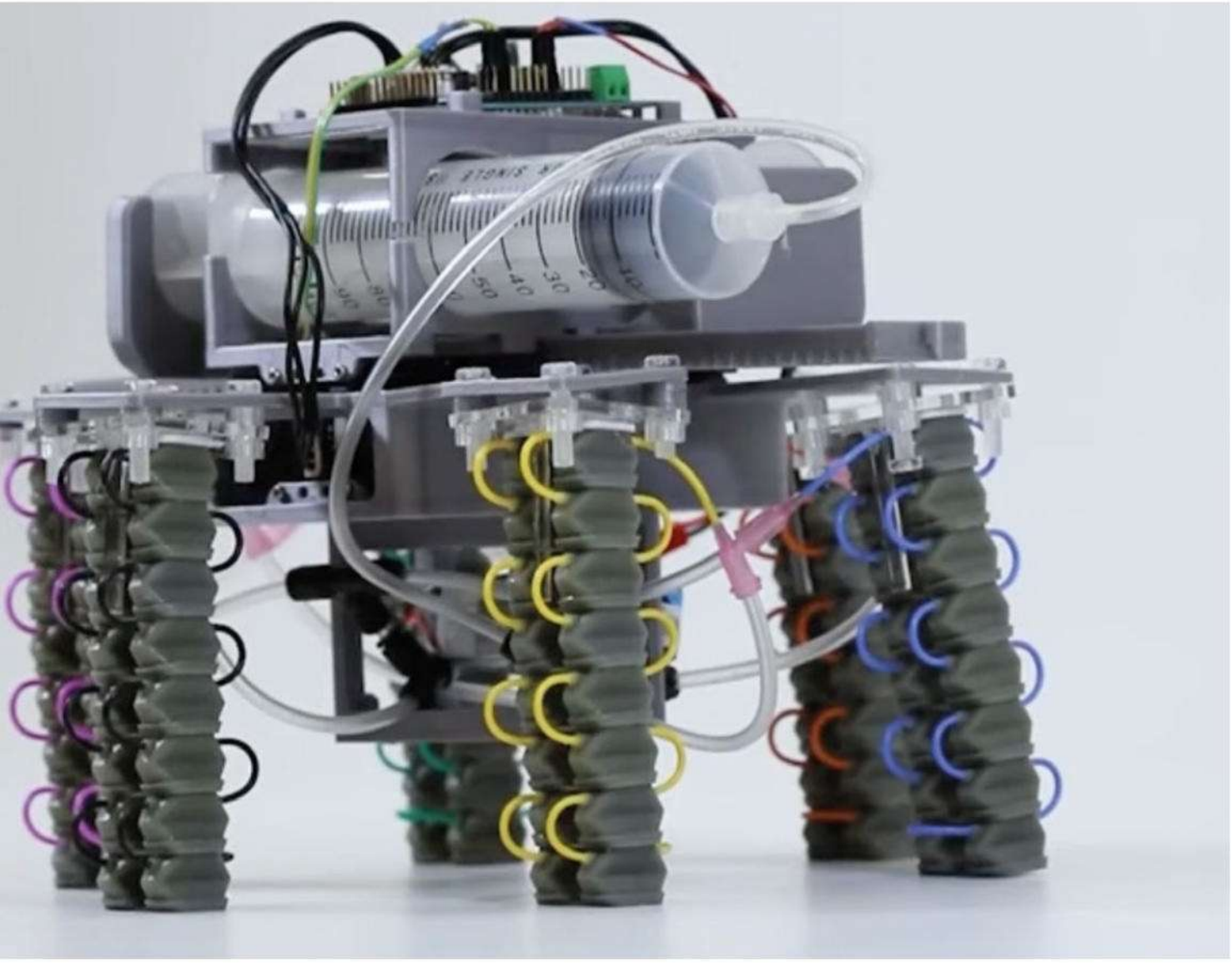
This robot, built by researchers at Cornell University's College of Engineering, uses a different approach. By using viscous fluid, which behaves unevenly, the team were able to get the actuators in the robotic legs to move more smoothly. The difference is akin to that between a jointed limb on a mammal and the movement of a caterpillar or worm; the clever thing is that by utilising the properties inherent in a viscous fluid, the hope is that robots using these actuators will need less computing power to perform quite complicated movements.



Right
This soft robot can walk at a speed of 0.05 body lengths per second, and can also crouch







Pico W Air

By **DPHacks** hsmag.cc/PicoAirQualityBoard

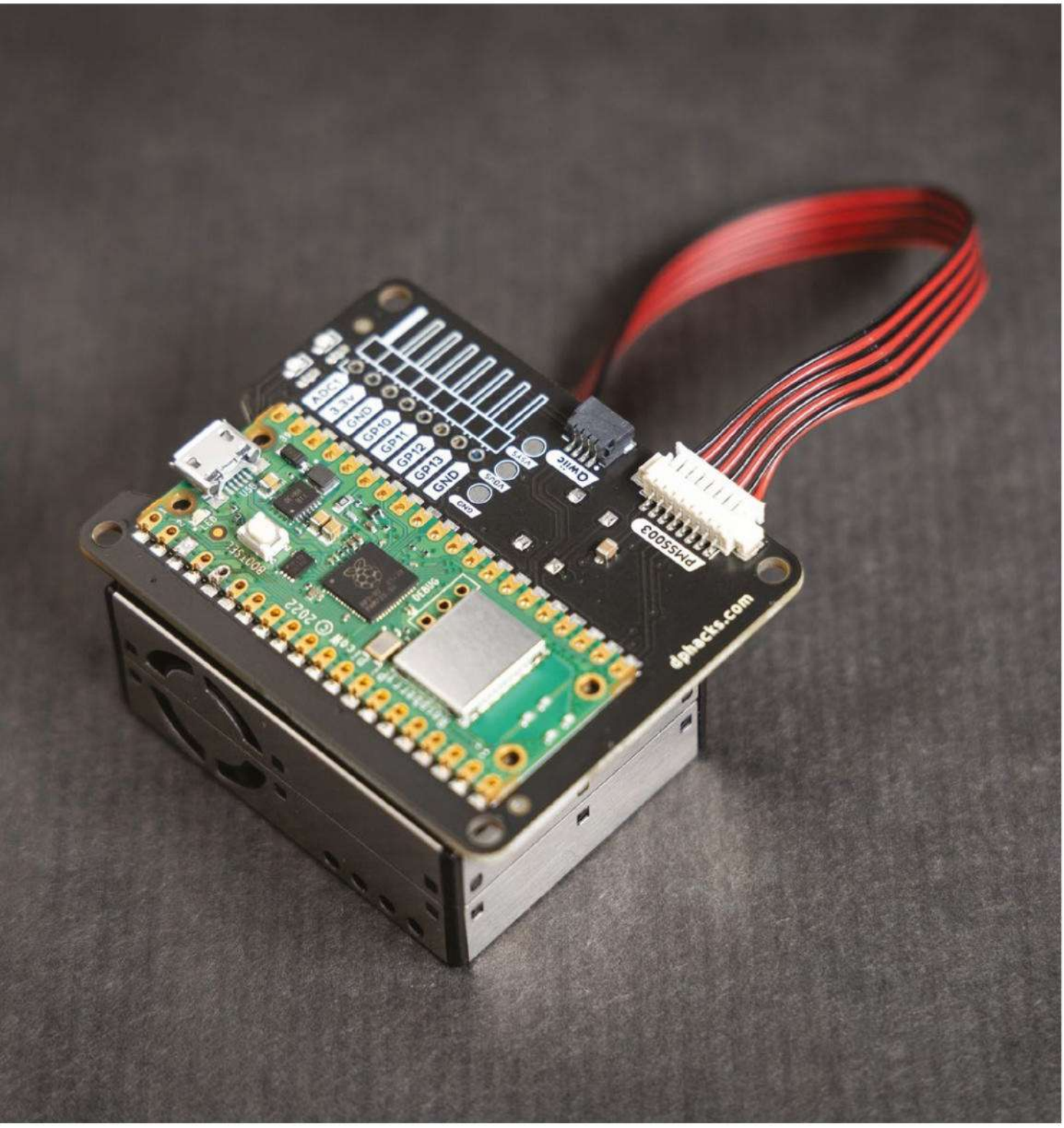
Free software and open hardware have enabled a boom in citizen science. This is just the latest device to add to our knowledge of the world around us.

This board comes with a Raspberry Pi Pico W, and adds a Qwiic connector for I2C devices, breaks out a few GPIO pins for 3.3V, ADC, GND, and four more pins. Most usefully though, is the built-in MQTT client and built-in HTML server, so it can easily transmit the environmental data coming in.

It comes with a connection for a PMS5003 particulate matter sensor, which you'll have to buy separately, along with a power source. Other than that, it's a one-stop device.



Right
We love a nicely labelled silkscreen


ADC1
3.3V
GND
GP10
GP11
GP12
GP13
GND
GND
SCL
SDA
DWIRE
PMS5003
dphacks.com
DEBUG
BOOTSEL
LED
2022

REGULAR

Star Wars Speeder Bike

By James Bruton hsmag.cc/StarWarsSpeederBike

YouTuber and inventor extraordinaire James Bruton has been building a bike based on a hoverboard. That's all you need to know really – go to the video right now and check it out. He's taken the electronics from the hoverboard, added a huge number of 3D-printed parts, a stem and handlebar from a bike, an Arduino Uno, some bearings and potentiometers, and has created a mode of transport that doesn't work – yet!

It's fascinating to watch the journey of experimentation that James is on, as he modifies one thing after the next to get a working prototype. The hoverboard motors, for example, at first struggled with the extra weight of all that plastic, so James just put higher-voltage batteries in. Trial and error has seldom been more entertaining.





Right
The bike uses omissions wheels; next time James is going to try mecanum wheels, last seen in HackSpace issue 71





REGULAR

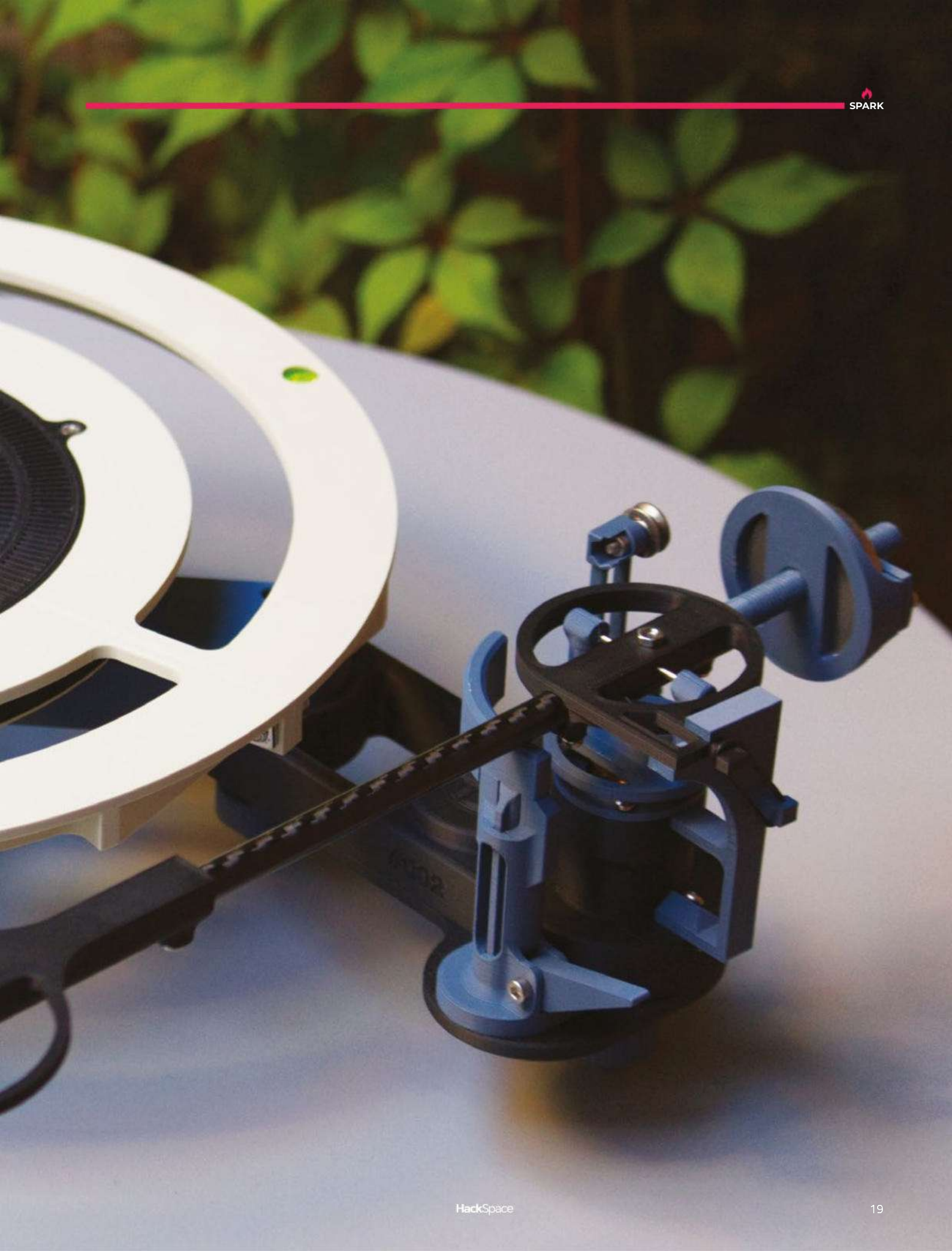
Objet 3d'art

3D-printed artwork to bring more beauty into your life



The world of high-end audio is baffling. There are a million expensive options, with a million ways of expensively buying the wrong thing. There's some absolute rubbish on the market – we shudder at the thought of a cheap steel stylus, rather than a diamond one, going anywhere near our original 12-inch copy of *Blue Monday* – and, at the other end, you can pay an arm and a leg to get the absolute best out of your pristine recording of John Cage's *4'33"*.

Because audio geeks love to experiment, upgrade, and tinker with things, we're surprised we haven't seen a 3D-printed turntable like this before. Designed by FilatoneInstruments, this turntable kit includes a lightweight carbon fibre tone-arm; built-in spirit levels, and stroboscope to ensure perfect levelling; brass bushings; a two-speed motor; and an Audio Technica AT3600L Cartridge (that's the bit that converts vibrations into electrical signals). Buy the kit from the Etsy store here: **hsmag.cc/FilaTone**, and download the STL files here: **hsmag.cc/FilaTone3D**.



Letters

ATTENTION ALL MAKERS!

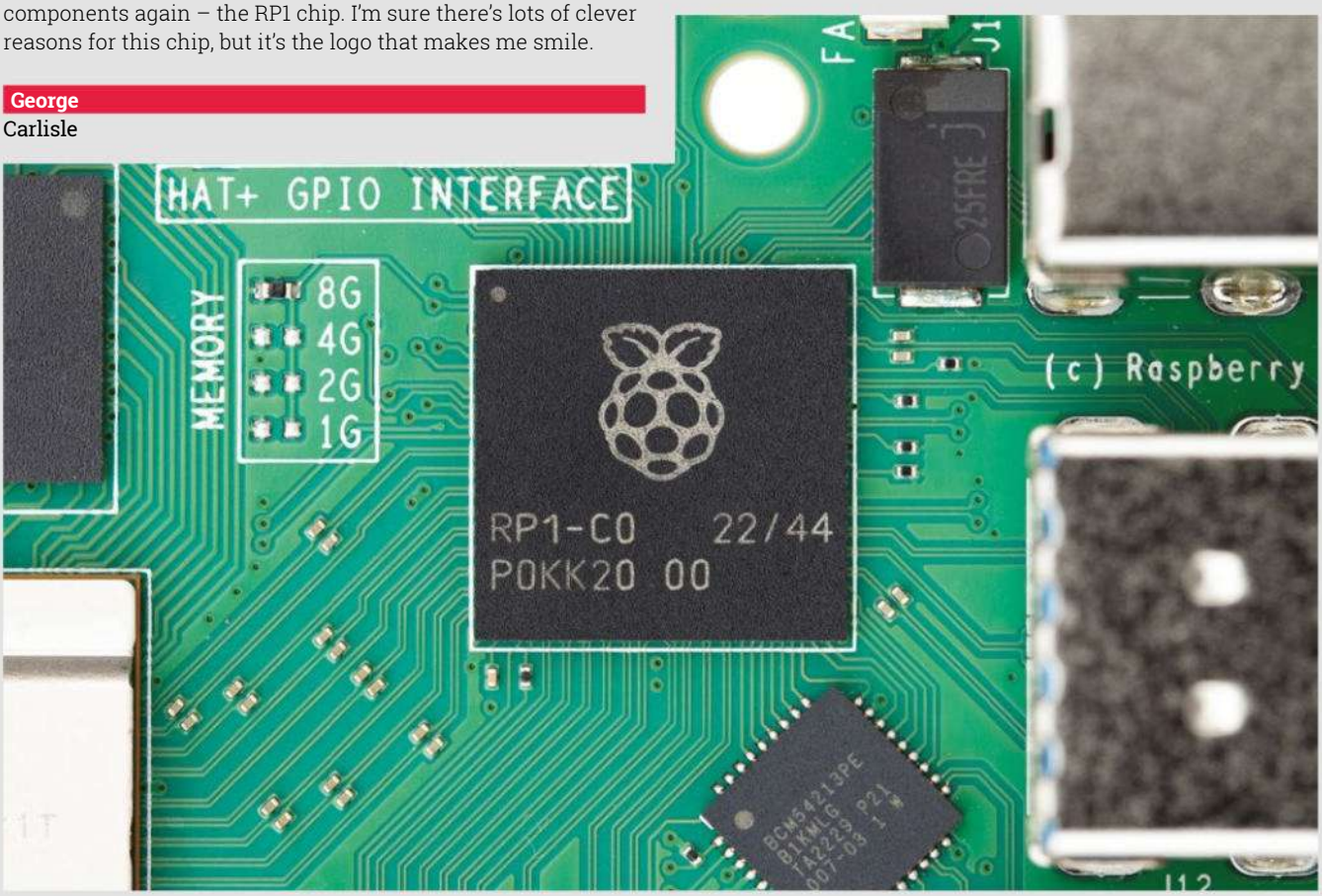
If you have something you'd like to get off your chest (or even throw a word of praise in our direction), let us know at **hsmag.cc/hello**

RP1

I was kind of sad when Raspberry Pi 4 launched. Don't get me wrong, it's a great computer, but it dropped the Raspberry Pi-embossed can on the Wi-Fi chips. I'm not sure why, but that just made me smile. While the Wi-Fi remains plain on the Raspberry Pi 5, I do like that there's a logo on one of the components again – the RP1 chip. I'm sure there's lots of clever reasons for this chip, but it's the logo that makes me smile.

George
Carlisle

Ben Says: It is an excellent logo, and deserves to be used more widely. I've put in a suggestion that, for Raspberry Pi 6, they arrange the components in the shape of the logo. It's going to take some pretty serious engineering work, but I think it's worth it.





SUBLIMATION

I've really enjoyed your articles on sublimation printing because I have ideas in my head for snarky T-shirts, mugs, and various paraphernalia that have – up until now – been stuck in my head. It's time to unleash them now. Brace yourselves for a torrent of snark, and it'll all be Andrew Lewis's fault.

Helen
Coventry

Ben Says: Honestly, sublimation printing is like magic. You print onto paper, then bam! You've got a mug or a T-shirt. It's also one of those techniques that impresses non-geeks. Custom mugs, and suchlike, make great gifts.

HEADPHONES

OK, we need to talk about those headphones. Prototyping is one thing, but those are horrific. Personally, I'd not be keen on having that battery there as well.

Sam
Bath

Ben Says: Yeah, I probably took the quick-and-dirty prototyping mindset a bit too far with those. That said, it did give me a valuable insight into what I want from a pair of headphones.

To be honest, I still haven't gotten over the loss of the headphone jack, but then the headphone jack itself could cause issues when it got stuffed full of fluff. I seem to have reached the point where I'm disappointed with both wired and wireless headphones, and it's just generally stressing me out a bit. Maybe I'm just hard to please.



We've got the new products your ideas deserve





We have over 400,000 new, name-brand products in stock and ready to ship–with more added daily. If you can design it, we can help you build it.

Find what you need at digikey.co.uk/new or call 0800 587 0991

DigiKey

we get technical

DigiKey is a franchised distributor for all supplier partners. New products added daily. DigiKey and DigiKey Electronics are registered trademarks of DigiKey Electronics in the U.S. and other countries. © 2023 DigiKey Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

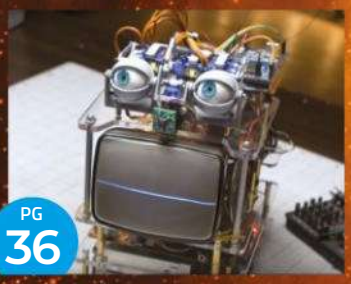
ECIA MEMBER
Supporting The Authorized Channel

HackSpace
TECHNOLOGY IN YOUR HANDS

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future



PG 36

HOW I MADE: ALEXATRON

A personal assistant with a little bit more personality

PG 44

INTERVIEW: JEFF GEERLING

Youtuber, developer, open source software advocate and maker of clever things

PG 24

Raspberry Pi Projects

Put the new computer to use



Raspberry Pi Projects

Put the new computer to use





The Raspberry Pi 5 will very shortly be available to purchase (and if you're a subscriber, you can pre-order from a guaranteed stock of Raspberry Pis, so you'll get one in the first batch. See hsmag.cc/prioritybording for details).

Its four cores are now the more powerful Arm Cortex-A76, which gives more processing power per clock cycle. The memory is now DDR4X, which gives faster access. Almost every form of connectivity is faster, and there are two new ways of connecting hardware: a UART port and a PCI Express port.

This all adds up to a much more powerful and capable machine. And, let's be honest, we geeks can be attracted to faster machines like moths to a flame. However, this issue, we're going to take a step back and look at some projects that can really benefit from the extra power this new board brings. Whether you're looking for a new desktop, building your own mini-data centre, or attempting to create sentient digital life.

But first, let's take a deeper look into exactly how the new computer is faster than its processors. →



FEATURE

Benchmarking

Getting to know the new computer

We've hammered the Raspberry Pi 5 with benchmarks, testing out just about every aspect of its performance, including 3D rendering, encryption, machine learning, and more. It should come as no surprise that it's faster than any previous Raspberry Pi, but by how much?

Last month, we interviewed Eben Upton and James Adams from Raspberry Pi Ltd, and they told us to expect a 2.5× increase in speed. Computer performance, however, isn't a single thing. Depending on your particular use, the bottleneck might be networking, disk access, CPU, or something else. We crunched a huge range of different processes to find out just how Raspberry Pi 5 stacks up against Raspberry Pi 400.

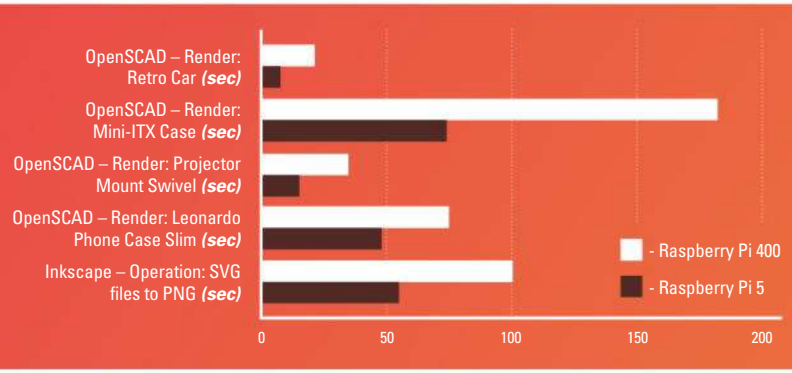
RUNNING YOUR OWN BENCHMARKS

You can compare the performance of Raspberry Pi 5 to your own computer by running our benchmarks and comparing them. We used the Phoronix Test Suite to hammer our computer in various different ways, and you can use this tool on your own hardware. It's available on Linux, macOS, and Windows. You can download it from **phoronix-test-suite.com**.

You can run either individual tests or suites of multiple tests one after the other. One problem we found was that many of the tests downloaded tens of gigabytes of data so, unless you have a truly stupendous microSD card, you'll need to clear out the installed test after each run.

CREATION

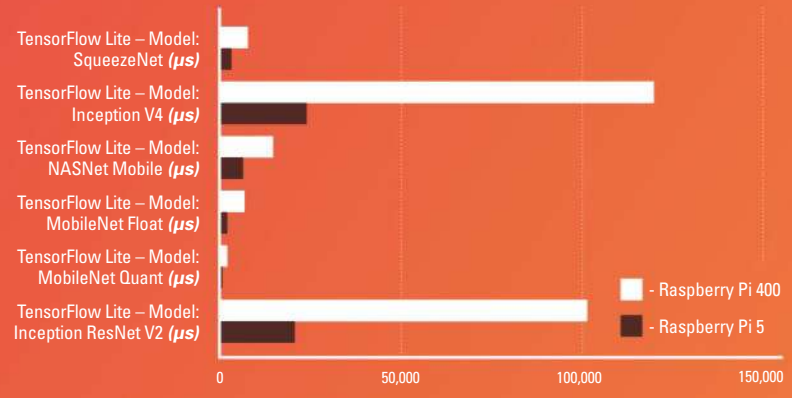
As makers, handling graphics involves some of the most computationally intensive things that we do. We put this to the test with both 2D and 3D graphics using Inkscape and OpenSCAD. As you can see, both areas got a significant speed boost, with 3D graphics seeing a bigger improvement.





MACHINE LEARNING

AI and machine learning are starting to creep into our lives in a huge number of ways. For benchmarking, we used TensorFlow Lite to do image recognition, partly because this is the most common form of AI we see in Raspberry Pi projects, but also because it is (in machine learning terms at least), well-established and understood. Here, we saw a greater-than-average performance increase (3.9×). So, it's a good sign that the Raspberry Pi 5 could be a great choice for embedded AI. This means either running the same models faster (or at higher resolution), or running more complex models.



SERVER

Many people run Raspberry Pis as little servers, either at home or for simple enterprise use. This can take a lot of different forms, but we tried to benchmark some of the more common uses. Results were a bit varied, but generally in the 2–3× range.





On any constrained device like a Raspberry Pi, there will always be trade-offs. What we've seen on Raspberry Pi 5 is a fairly broad improvement in performance across the board.

There's faster storage, faster connectivity, faster memory, and a faster processor. All of those have kicked up in performance and the 2.5× performance improvement Eben Upton promised us seems about right. That said, there are some areas where there's a significantly higher uptick in performance, and that's probably due to the increased processor cache and NEON.

It's worth bearing in mind that this is likely the slowest Raspberry Pi 5 will run. As we've seen with previous models, future software improvements are likely to give performance improvements. →

> There's faster storage, faster connectivity, faster memory, and a faster processor

BOOKWORM

Alongside Raspberry Pi 5, there's a new version of Raspberry Pi OS, known as Bookworm. This is based on Debian Linux, and the new version brings in a whole host of improvements and bug fixes. The most important for our benchmarking is the upgrade to Python 3.11. This brings a fairly significant performance boost of around 60%.

Obviously, this doesn't just apply to Raspberry Pi 5, so you can get a nice little speed-up for any Python scripts by grabbing the latest OS and flashing it to your SD card.

Machine learning

A brain that can see

We've seen from the benchmarking that Raspberry Pi 5 performs particularly well in machine learning using TensorFlow Lite. Well, machine learning is perhaps a bit of a misnomer there because we didn't test the learning part (which is a complicated and specialist task), but what to do once it has learned.

TensorFlow is a tool for running neural networks. These are computer programs structured in a way inspired by the way human (and other animal) brains work, with a group of neurons that react differently to different stimuli. You don't program these models, but train them – this is the process of showing them a lot of known input. In the case of image classification models, this means showing them a lot of images, and also feeding in information about the contents. Each image adjusts the model slightly, making it more accurate.

Once you have a trained model, you can then run it on new images and it will attempt to recognise things in them. It's this process that we can run on a Raspberry Pi.

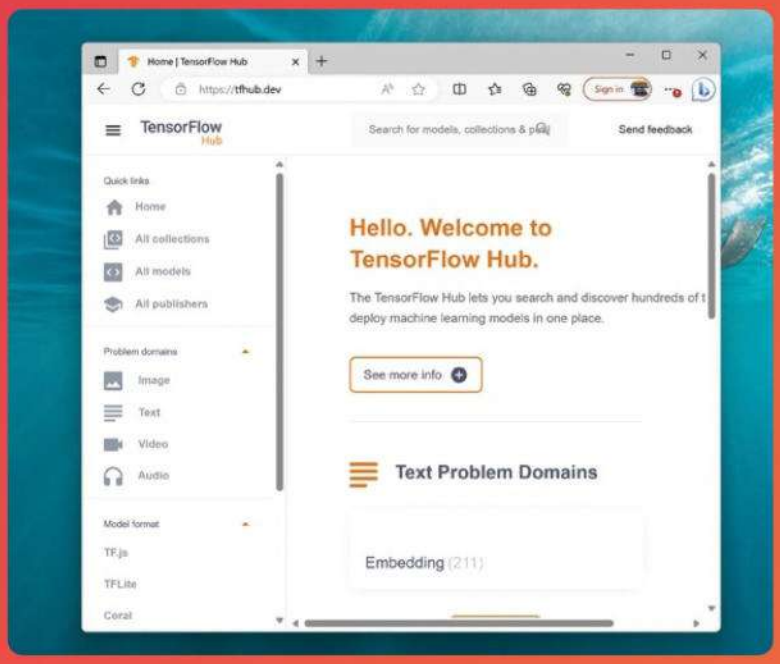
Models vary hugely in size and complexity. Some require specialist hardware with huge amounts of memory to run, and some can run on tiny microcontrollers.

The most common tool for running these models on Raspberry Pi computers is TensorFlow Lite. This can be used with a few different programming languages, but the most common is Python. There is a slight problem that, at the time of writing, the main Python package for TensorFlow Lite doesn't support the version of Python in the latest 'Bookworm' version of Raspberry Pi OS, but we can grab an early version that does.

This is available at **hsmag.cc/TFLitePy**.

Specifically, the version we need is:

```
tflite_runtime_
nightly-2.14.0.dev20230612-cp311-cp311-
manylinux2014_aarch64.whl
```



Below
You can download pre-trained models from TFHub.dev



Above
The MobileNet recognises many common objects, such as a banana



Left Raspberry Pi 5 has two ports that can be configured as either camera or display. You can use TensorFlow Lite with either of them

You can install this with:

```
sudo pip3 tflite_runtime_
nightly-2.14.0.dev20230612-cp311-cp311-
manylinux2014_aarch64.whl -break-system-packages
```

Once you have that installed, you can use TensorFlow Lite in your Python programs.

We tested this out with the example code from the Raspberry Pi blog (**hsmag.cc/tflite**). This code uses the Picamera2 library to bring in images from camera modules and feed them into a TensorFlow Lite model.

This worked, but didn't handle autofocus on the Camera Module V3 very well. You can grab updated code that continuously autofocuses from **hsmag.cc/tflitewutofocus**. This runs in exactly the same way as the code from the blog (and you'll need the data files from the repository for it to run).

> The most common tool for running these models on Raspberry Pi computers is TensorFlow Lite

OVER TO YOU

We're really excited to see what people do with this. To whet your appetite, here are a few projects built on earlier versions of Raspberry Pi that could get a performance boost from the new version.

- A self-driving car (**hsmag.cc/aicar**). While full-sized self-driving cars are slowly getting onto our roads, this one is just a model to test out the technology. This car, created by YouTube user Hungry Man, uses TensorFlow to detect things like stop signs.
- An AI weather station (**hsmag.cc/airai**). Kutluhan Akar trained a TensorFlow Lite model to try and predict air quality based on sensor readings. In most TensorFlow projects, Raspberry Pi is used to run the models, but in this project, Raspberry Pi is used to train a project that then runs on an Arduino.
- Smart home AI camera (**hsmag.cc/homeai**). There are many smart home integrations that you can do based on image recognition: turn on the heating when your car pulls into the drive, or add items to your shopping list when your fruit bowl is empty, for example. This Home Assistant add-on helps you use TensorFlow models to interact with your other smart gadgets. →

FEATURE

DIY computer

Create your own custom-built machine

With more computing power than ever before, Raspberry Pi 5 makes a great device for building your own computer with. Obviously, this can be as simple as putting it on your desk and plugging in a keyboard, laptop, and monitor, but the adaptability of Raspberry Pi means that you can create something far more complex if you'd like.

Way back in issue 24, we built an unconventional portable computer with a Raspberry Pi 4. It's now time to modernise it with a brand new Raspberry Pi 5.

Below
It's a bit bulky, but solid, and very hackable



Before we look at the parts you need to build a computer, let's think about why you want to. You can go to a shop and buy a computer, and there are literally thousands to choose from. However, despite there being a huge number of models, there really isn't much choice. In reality, you can get three different computers. You can get a big box to sit on your desk, you can get a foldy one with a screen and keyboard attached, or you can get a handheld rectangle with no keyboard. These three come in a range of specifications with a few software choices, but that's basically it.

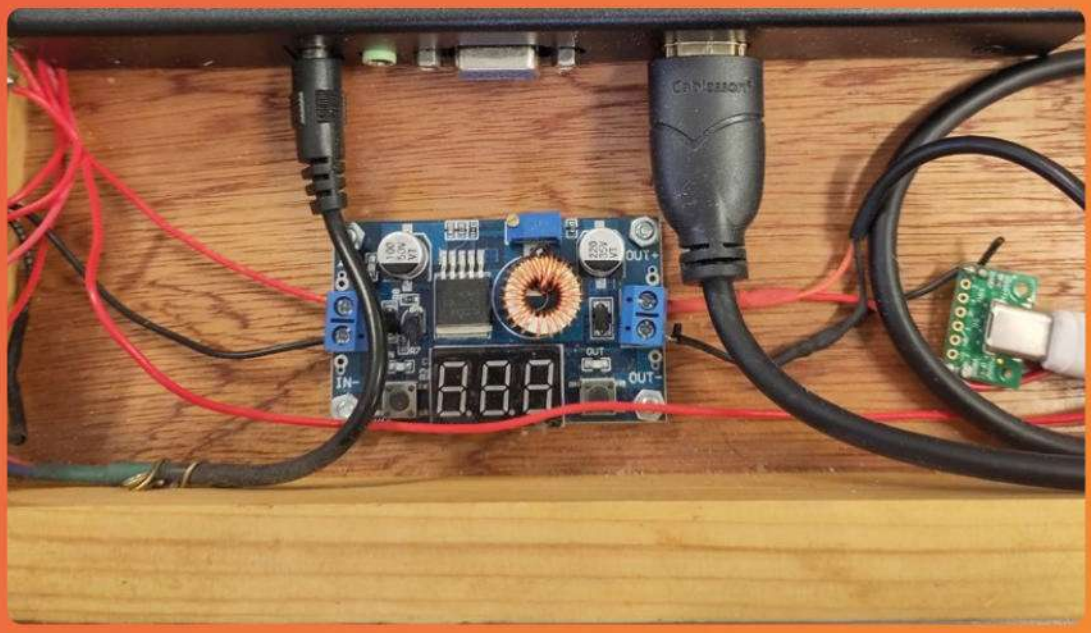
Building your own lets you throw away this tri-opoly and start with a blank slate. That's exactly what we did.

Our main requirements were that we lifted the screen up to make it more ergonomic, that it protected what it was carrying, and that there was space for hacking. We used it as a main computer for a while, but it did eventually end up in storage under a bed.

> It's nice to have something to give us a little connection to the natural world

Now it's time to give it a refresh. The first thing we can say is that we're really glad we made it out of wood. This material is almost endlessly hackable. We've pulled some bits out, added some new ones in, and generally gave it a bit of a shuffle around. We also really like the feel of wood. In a world of plastic and aluminium, it's nice to have something to give us a little connection to the natural world.

However you choose to build your computer, there are a few things you'll need.



Left
This power module gives us 3A at 5(ish) volts

Below
We're playing with the options for 3D stereoscopic photography at the moment, but haven't settled on a camera setup, so it's just taped together for now

POWER

Obviously, we need 5V of power for the Raspberry Pi. Ideally, we should be able to supply 5A of power, but we don't really need this much and we can get away with closer to 3A. You'll also need power for whatever else you have on the board. We need 12V for a screen, so we've gone with a 12V power supply and a switching regulator to take this down to 5V for the Raspberry Pi, which we feed in via USB-C.

SCREENS

You can connect screens to either the Camera / Display ports or the HDMI ports. The Camera / Display ports allow slightly tidier wiring, but there are fewer display options. Using HDMI, you need cables with quite long connectors (and the cables themselves tend to be quite long).

In our original build, we opted to go with HDMI connectors, though it is the least tidy part of the build as we have coiled HDMI cables on show. For now, we're going to stick with this option, though we will stay on the lookout for alternative options.

Originally, we had two screens, but we found we didn't use the smaller of the two very much, so we've removed that to give us more space for alternative hardware. This also removes some of the untidy HDMI cables.

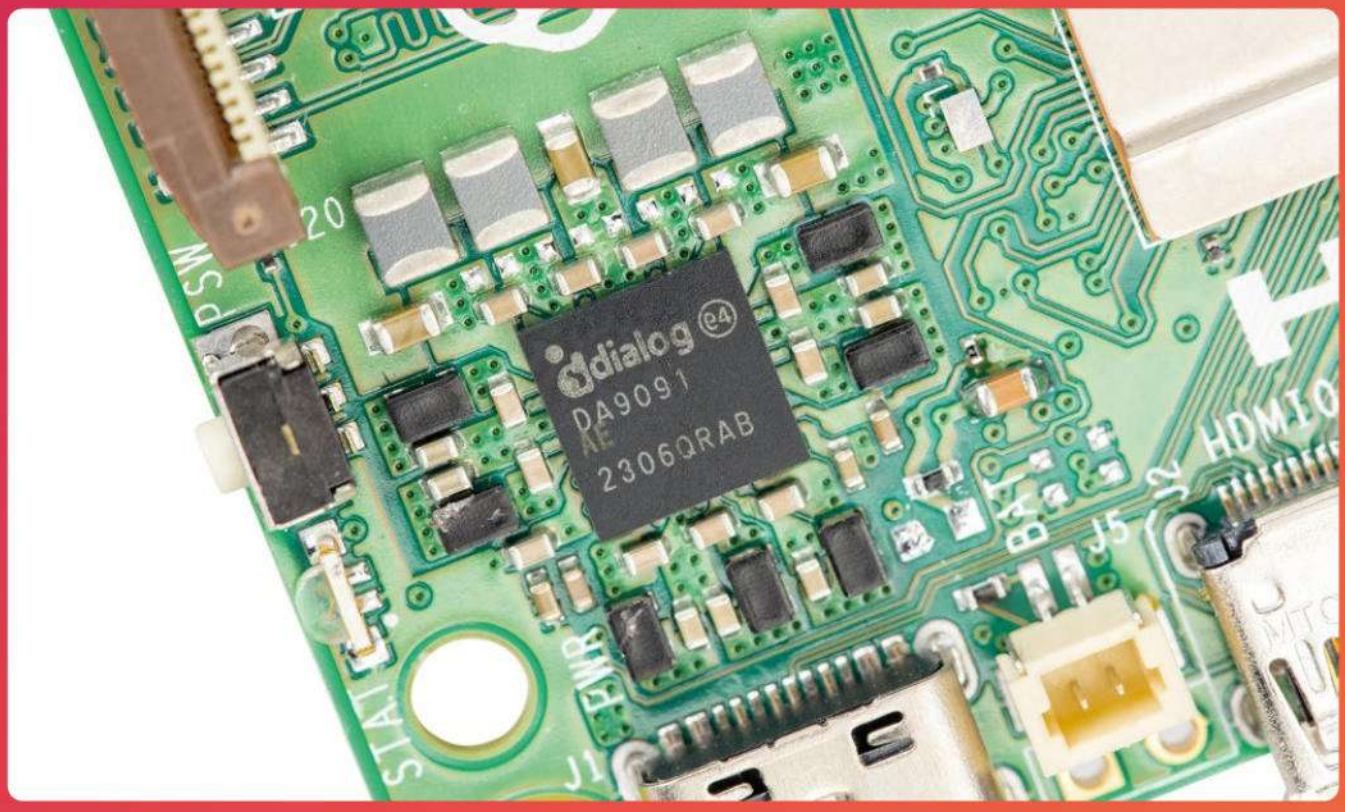


KEYBOARD

Our portable computer doesn't have a keyboard or mouse attached. However, you just need to plug in your own (or use a Bluetooth one that's not plugged in). For the authentic Raspberry Pi experience, we've opted to use an official Raspberry Pi mouse and keyboard. They're tough, reasonably priced, and look the part.

You can combine these elements however you like to make a computer that suits you perfectly. →

FEATURE



Home server

Build your own tiny data centre

As well as a faster processor, Raspberry Pi 5 can get data in and out a whole lot faster thanks to a PCIe port, faster USB ports, and faster SD card access. This means that, while the Ethernet port and Wi-Fi are still technically the same speed as before, you may well find yourself with a speed boost when using the new board.

There are loads of uses for a home server, but here are a few of our favourites:

NEXTCLOUD

Cloud computing has come to mean a lot of different things over the years. These days, it's commonly used to refer to user-focused apps that often run in a web browser for things like word processing, calendar, and file storage.

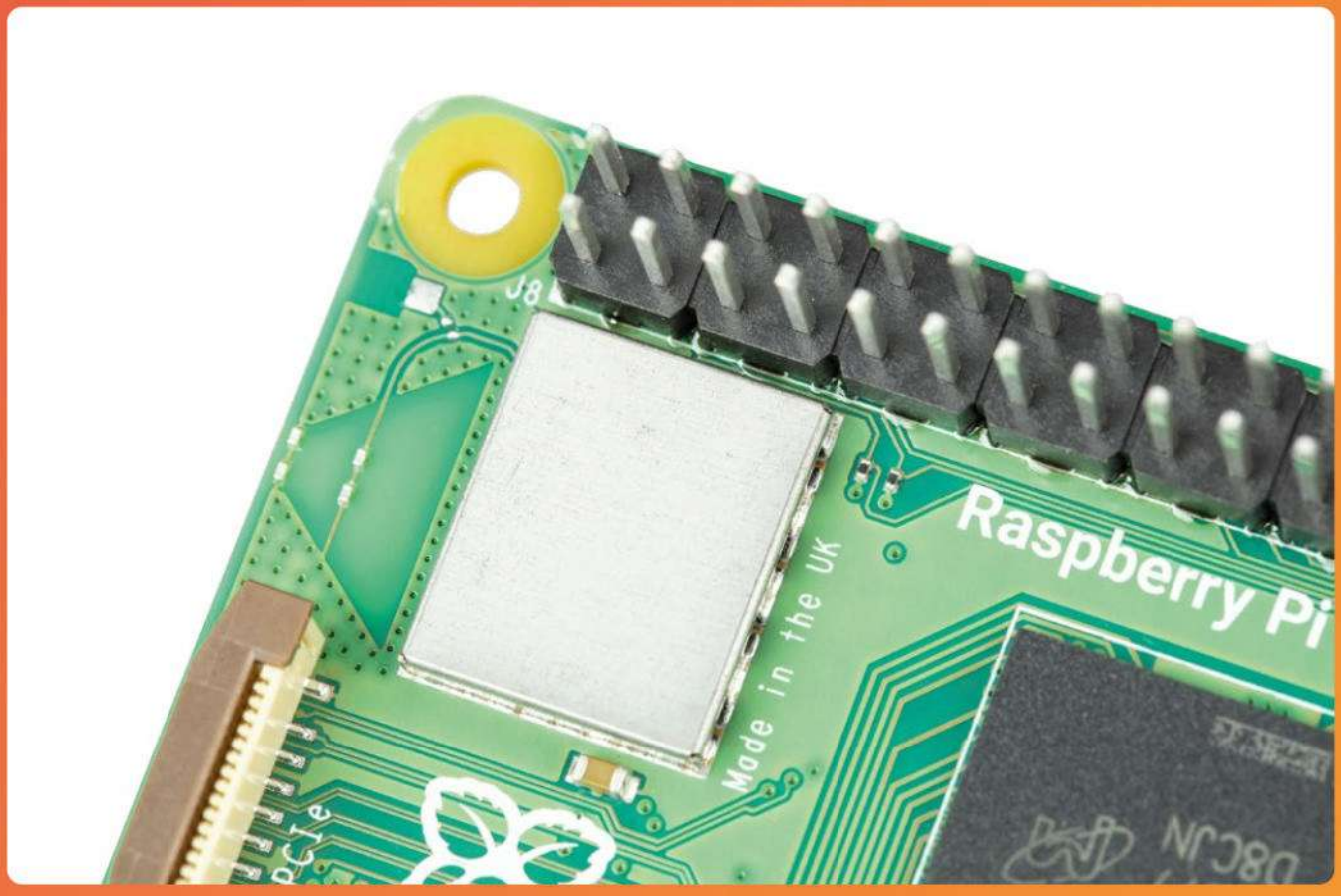
While the online clouds can be useful, they also have issues. There are often limitations on use (such as the amount of data you can store), and they are run by companies that may have policies you disagree with. Nextcloud lets you run a host of popular cloud computing services on any hardware you like, including a Raspberry Pi. You can edit your documents in a web browser, share files, and much more. Because it's running on your hardware, you're in control.

LIBREELEC

Remember when you could go out and buy media? A DVD of your favourite film or TV series? Now it seems that there is only streaming services, with a motley collection of shows that are prone to disappearing at the whims of licensing agencies.

It doesn't have to be like this. You can have the benefits of owning media and the benefits of streaming media with LibreELEC. This software lets you stream videos that you own to your TV or other devices.

Above
Double-tap the power button to shut down gracefully. This lets you safely power down without a keyboard attached



Above
While the networking hardware remains the same, the improvements around it mean that you might get faster networking speeds

HOME ASSISTANT

A smart home is a fancy term for having internet-connected devices that can be controlled from a central location. This could be an intelligent lighting system, a heating system you can switch on before you get home, a security system, and much more. There are loads of options for these smart devices, from off-the-shelf appliances you can get from your local electronics store to homemade builds.

The key part of a smart home is the central hub that brings it all together. This should be able to talk to the devices and also let you control them in the way you want to. Not all hubs can talk to all devices, so getting the right hub is a critical part of building the smart home system you need.

Home Assistant runs on a Raspberry Pi, can talk to a lot of different bits of hardware, and gives you powerful tools for controlling your home. There's an easy-to-use graphical interface, or you can dive in and write scripts to automate common bits of hardware control.

PRIORITY BOARDING

Right now, the factory in Wales is churning out Raspberry Pi 5s as fast as possible. However, there's almost certainly going to be a huge demand for these little computers and, by the time you read this, if you haven't already got your order in, there's going to be a long wait.

However, we've secured enough Raspberry Pi 5s to allow every subscriber to HackSpace magazine (and our sister magazine, The MagPi) to buy a Raspberry Pi 5 at the recommended retail price. These will be available for pre-order before launch, and are guaranteed be in stock to ship at launch.

If you're an existing subscriber, just sit back and wait for an email with the details. If you'd like to get this fantastic offer, subscribe at **hsmag.cc/subscribe**.

Six-month subscriptions start from just £30 / $43 / €43 for UK / USA / EU. You'll get six magazines delivered to your door for less than the cover price, get a free Raspberry Pi Pico W, and get the exclusive chance to order a Raspberry Pi 5 at launch without them selling out. See page 62 for more details.

SUBSCRIBE TODAY



GET SIX ISSUES FOR JUST:

£30 UK / **€43** EU / **$43** USA & Canada

GUARANTEED
RASPBERRY PI 5 AVAILABLE AT LAUNCH
for all subscribers



SUBSCRIBER BENEFITS:

- **Get every issue of HackSpace magazine delivered to your door**
- Beat the crowds with guaranteed Raspberry Pi 5 stock for subscribers
- Early access to the PDF edition
- **Get a free Raspberry Pi Pico W**

hsmag.cc/subscribe

Subscribers will get a voucher giving them the chance to purchase one Raspberry Pi 5 from reserved stock at The Pi Hut (thepihut.com) for full retail price. Reserved stock means that these will be available even if they are out of stock for general purchase.

FEATURE

HOW I MADE

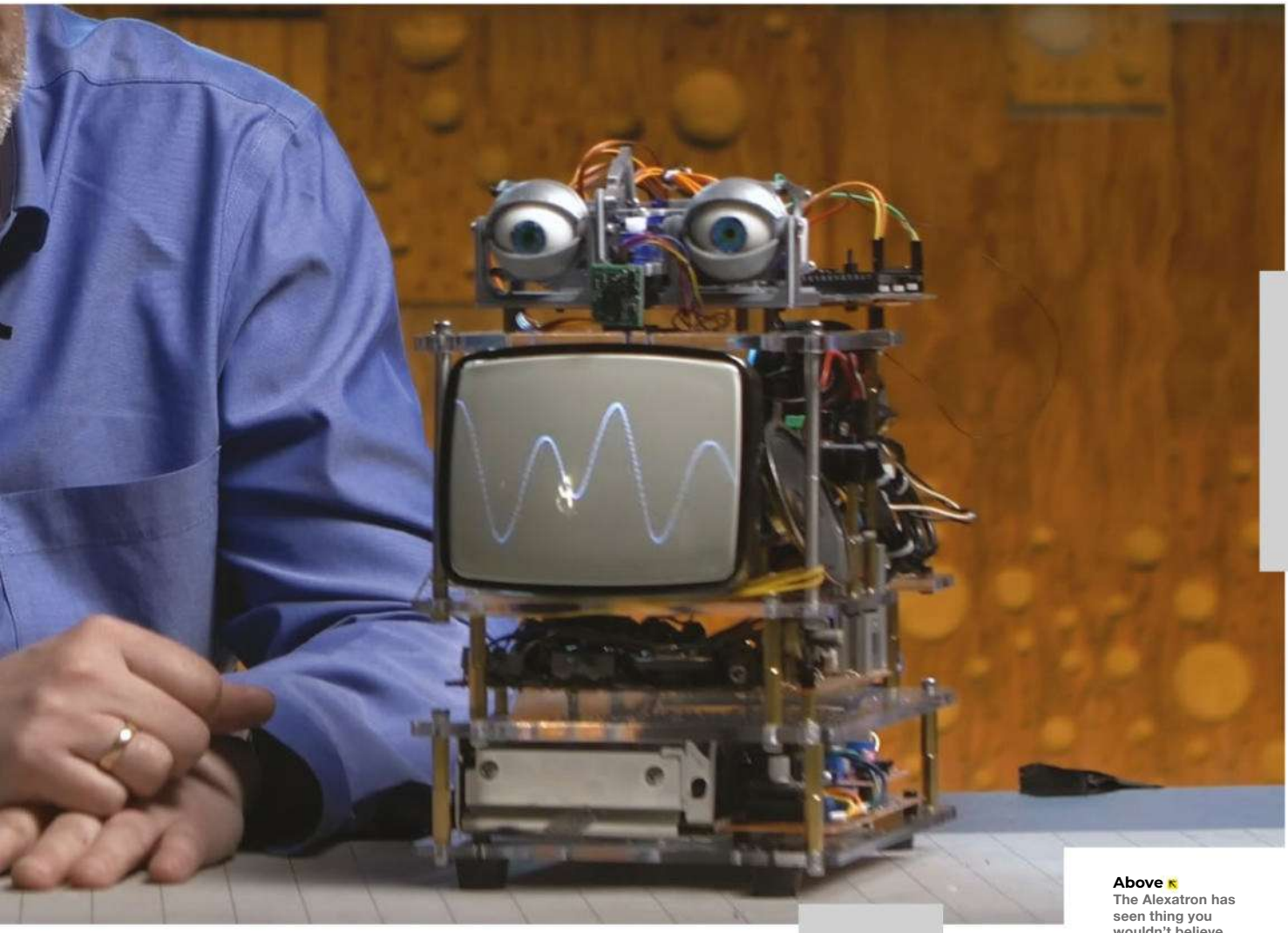
By Thomas Burns



ALEXATRON

An Amazon Alexa with a little bit more personality

Give a man a standard home assistant and he'll ask it the weather forecast, the football results, and whether the traffic's likely to be bad on the commute into work. But give that home assistant a pair of animatronic eyes and a CRT mouth, and suddenly you've got a robotic person. Kind of. That's what Thomas Burns did with his Alexatron. Thomas reckons it's the coolest thing he's ever built, but don't take his word for it: watch his build video at **hsmag.cc/Alexatron**, get excited about building your own, and wonder why more robots can't be a little bit more human.



Above
The Alexatron has seen thing you wouldn't believe

HackSpace: What was the inspiration for the Alexatron? Why did you feel the need to make an Alexa more human?

Thomas Burns: As a kid in the 1980s, I was enthralled with the vision of technology that we often saw on TV and in the movies. I'm talking about Johnny 5 and R2-D2 and K9. These guys weren't just computers, they were computer creatures. But today, the most interactive tech in my house – an Amazon Alexa – doesn't look anything like that. It's basically just a small black hockey puck. Alexa is great, but whenever I use it I find myself asking, what happened?! So the goal here was to breathe some life into an Alexa to try and make it feel a bit more like the kind of electronic companion I'd always dreamed of having in my home. Or even better, at the workbench!

There's a ton of R&D going into creating human-like robots these days: human voices, human heads and faces, etc. But I feel like the closer we get to humanoid, the more it disturbs people. What I want, and what I think many users want, is a creature much more like a Muppet, or a pet. A harmless, curious creature with a bit of personality. Something like the animated toys that greet JF Sebastian when he comes home to his apartment in the film *Blade Runner*. So, the Alexatron project was basically a proof of concept for the kind of creatures I'm focused on making these days.

HS Where did you source the screen? LEDs are everywhere nowadays, but CRTs not so much.

FEATURE



Above ↑
Thomas broke out the power from the Alexa to trigger the CRT

TB My goal was to inspire the user to forget, even if only for a few seconds, that they are talking with a computer. So I needed a mechanism for visualising the creature's voice that didn't appear digital. Using a waveform on a CRT has a much more organic feel than an LCD, in part because the waveform feels unique every time the creature speaks. There's also the nostalgia factor of a CRT, which helps elicit an emotional connection from the user, something I believe is critical to achieve truly meaningful human-computer interaction.

"I DISCOVERED ELECTRONICS RELATIVELY LATE"

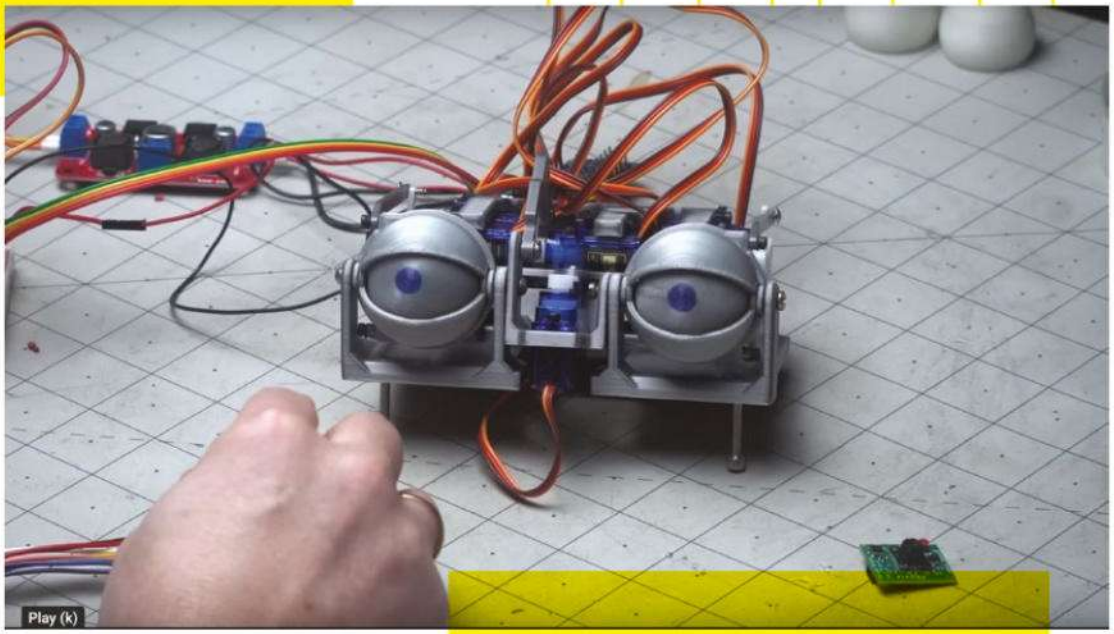
But, unfortunately, the CRT isn't a viable option for future iterations of this build due to its size, and also because the high voltage required to run it (around 8000 volts!) is a bit of a safety concern. So now I'm looking at ways to recreate a unique waveform 'mouth' on an LCD or OLED screen. The trick is making this digital presentation feel more analogue.

The Samgori Market is one of Tbilisi's many gems, and a place that I visited regularly for inspiration. On dark, rainy days it can feel very much like a scene from a science-fiction film, which is exactly the kind of place where I want to search for parts. I love using forgotten analogue technology in my builds, and several of Tbilisi's markets are awash in Soviet tech, which is perfect for my style. I hope that place never changes!

HS You obviously know your way around a TV circuit board, and the guts of the Alexa. Are you an electrical engineer by trade?

TB I discovered electronics relatively late in life, but when I finally did discover it, it felt like the door opened to an entirely new creative universe. One of my first projects was using parts from an old Soviet black-and-white CRT television to make an adjustable DC power supply for my workbench. It's a bit clunky and not exactly what I would call 'stable', but that feeling of turning it on after so much work and seeing it come to life is truly indescribable. I remember in those early months I would start tinkering at my workbench once the kids went to bed, then get lost in the flow, and suddenly realise it was already 05:00 in the morning.

I'm 100% self-taught, and I'm very grateful that so much knowledge is available



Left
The animatronics were inspired by Will Cogley – check out his Youtube channel here: www.youtube.com/c/WillCogley

at our fingertips these days. There's never been a better time to be a maker!

HS And the animatronics: is that your own design, or are you standing on the shoulders of giants?

TB Everything I've ever made was done by standing on the shoulders of giants. The animatronic eyes are an open-source creation by the talented Will Cogley, who has a wonderful YouTube channel with some very impressive builds. I'm considering designing my own animatronics module for future builds, but for the Alexatron project it offered a fast, simple way to get things up and running.

HS How did you handle the face tracking to determine where the eyes are looking?

TB I decided early in the development phase that eye contact would be critical in reaching my goals for the project. So I did a ton of research into how to achieve that. At one point, I had come to the discouraging conclusion that I'd need to spend dozens of hours programming my own face-tracking algorithm on a Raspberry Pi. But then I stumbled onto the Person Sensor, a tiny face-tracking camera board made by a wonderful start-up called Useful Sensors. I paired the Person Sensor to an Arduino microcontroller, and was able to use it to drive the animatronic eyes so that they follow the user. I had to make a few modifications to the code to make it work for my project, but it's really a wonderful bit of kit.

HS Did anything go wrong while you were working on the Alexatron?

TB Are you kidding? Everything went wrong! But that's what prototyping looks like: it's an incredibly messy process. When you're trying to get to a proof of concept, the best strategy is to move fast and break things. Here's a few of the hundreds of problems I encountered along the way:

- My initial power supply wasn't powerful enough to smoothly drive the servo motors, but more powerful supplies were too large for the chassis;
- After soldering wires to the Amazon Alexa's daughterboard to monitor voltage fluctuations, I discovered that the Alexa wouldn't work correctly outside its case (due to a grounding issue, I believe), which required a complete redesign of part of the chassis;
- I built an entire part of the chassis structure around an Arduino Nano,

FEATURE



Above
It's watching us, as well as listening to us

only to discover that the voltages I was using were destroying all the Nanos I installed;

- I used heat to get the bubbles out of the glossy resin on the irises of the eyes, only to later discover that the heat had also warped the eyeball shape;
- During the final assembly push, I discovered the only microcontroller remaining in my kit required a different type of cable than I'd been using, so I had to drive all over Tbilisi, on a Sunday when everything was closed, trying to find the right cable.

The list goes on and on and on! I had to constantly remind myself that if I wasn't encountering challenges along the way, I probably wasn't reaching far enough.

HS Do you find yourself using it more/ interacting with it differently now that it's not a bog-standard Alexa?

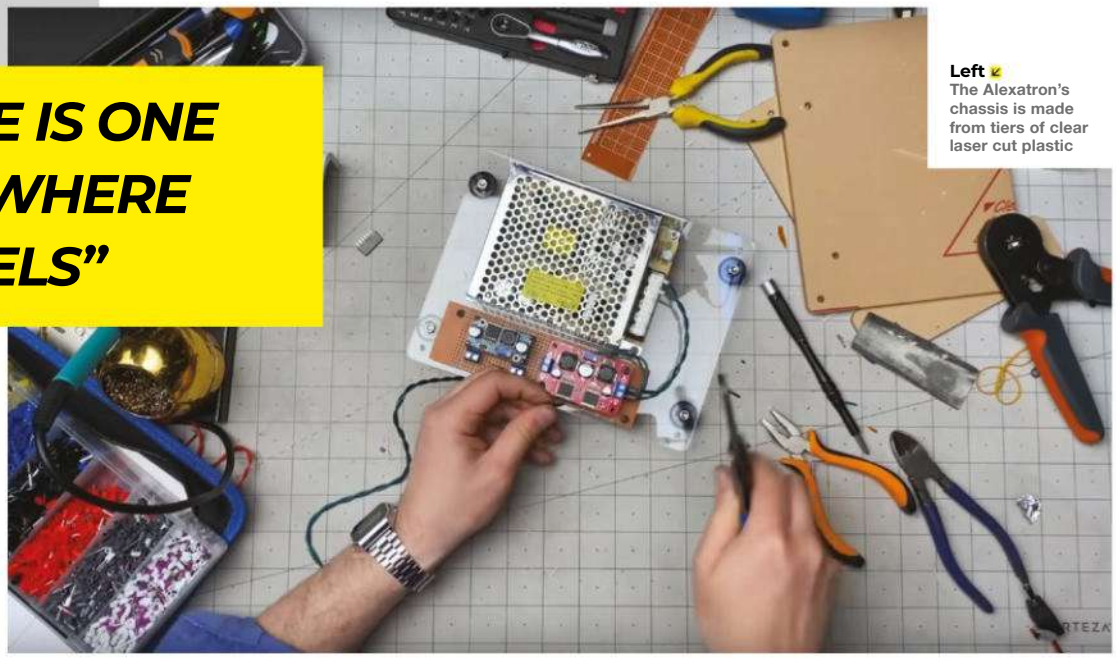
TB I've been using the finished build mostly as a research tool, observing how users interact with it. And the most valuable data comes from watching how children use it, because in many ways it's a much more pure interaction, unencumbered by a lifetime of working with technology. The best feedback I've gotten to date has been from parents talking about how their children reacted to the build video!

HS Did you have to do anything special with the Alexa's settings to make it quote *Blade Runner* to you, or has it learned that from your search history?

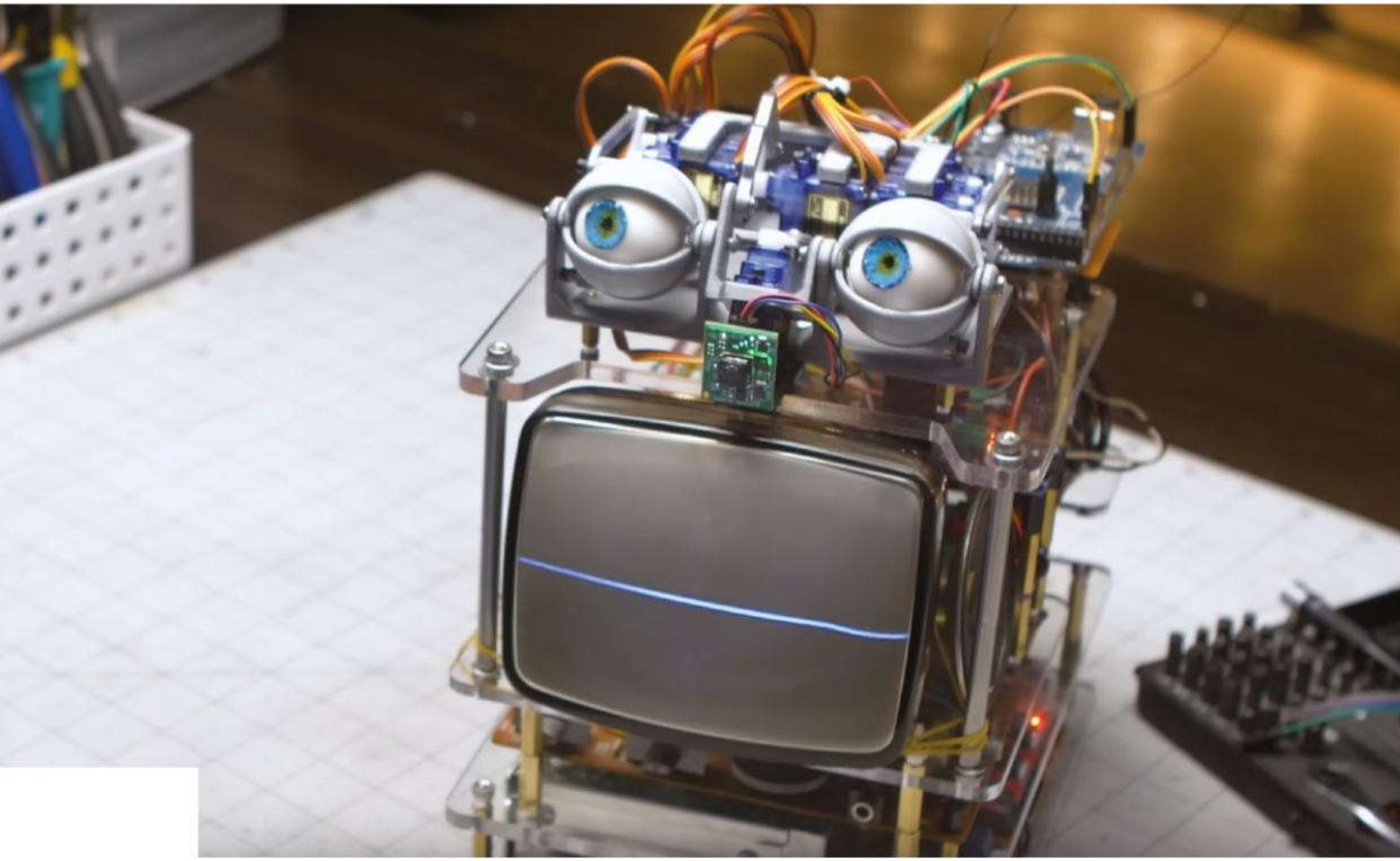
TB The Alexa platform does offer some limited options for customisation. You can change the wake word (from a short list of options), you can choose a voice from their list, and you can also program answers to specific questions. By the time I reached the final stages of the build, I already knew that the next iteration would use an LLM or AI platform instead of the Amazon Alexa, so I wanted to see how it would respond in more creative ways.

Even though the Alexa is in many ways very limited as a 'brain' for an interactive

"THERE IS ONE AREA WHERE IT EXCELS"



Left
The Alexatron's chassis is made from tiers of clear laser cut plastic



Above
The component roughly where the robot's nose should be a camera with built-in facial recognition processing, made by Useful Sensors

build, there is one area where it excels: response time. Alexa's response time is fast enough so that conversations with it don't feel laboured. At the time of this writing, all of the speech-to-text and text-to-speech options available for AI and LLMs suffer from very high latency, where responses require as much as four to five seconds before being delivered. I'm sure this will change in the coming months, but for now it's a big hurdle for me in moving this project to a different platform for interactivity.

HS What's next for version 2?

TB Moving it to an AI platform will be the big change. But AI is changing so incredibly fast these days: by the time I decide on a direction forward, some new tech will inevitably come out that changes everything. ChatGPT, for example, was released just after I entered the production phase on the Alexatron build. Do I stop everything and start over with ChatGPT? Well, no. Because if you constantly do that, you'll never get the project across the finish line.

Version II of this build will be designed to help move the project toward something that is smaller, lighter, more interactive, and more accessible to makers. I'm not sure if it will be a kit that makers can use to assemble and customise their own versions, or possibly even a dedicated product line of toys, but I think there's a lot of potential here for a production run. I've never done a production run of anything before, so this is all new to me. Right now I'm still in the early tinkering stage, where it's just me and a lot of late nights in the workshop. Looking forward, however, I'd like to be able to build a small, interdisciplinary team to scale this project. I am often reminded that the best innovation usually happens at the intersection of disciplines, and I get goosebumps thinking about all the possibilities!


FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
NEW 2022 UPDATE
RETRO GAMING WITH RASPBERRY PI
2ND EDITION
164 PAGES OF VIDEO GAME PROJECTS
+ BUILD AN ARCADE MACHINE
PLAY & CODE GAMES!

RETRO GAMING WITH RASPBERRY PI

2ND EDITION

Retro Gaming with Raspberry Pi shows you how to set up a Raspberry Pi to play classic games. Build your own games console or full-size arcade cabinet, install emulation software and download classic arcade games with our step-by-step guides. Want to make games? Learn how to code your own with Python and Pygame Zero.

- *Set up Raspberry Pi for retro gaming*
- *Emulate classic computers and consoles*
- *Learn to code your own retro-style games*
- *Build a console, handheld, and full-size arcade machine*



BUY ONLINE: **magpi.cc/store**



HackSpace magazine meets...

Jeff Geerling

A man who makes things happen – it's Jeff Geerling everyone!

If you want to cut through the buzzwords and find out what tech actually does, you could do a lot worse than listen to Jeff Geerling. He's built an utterly wholesome corner of the internet in which he explains, demonstrates, and demystifies the latest thing that everybody else pretends to know about. Linux, single-board computers, open source, developer tools, home automation – Jeff does the lot, and he wants to teach you how to do it, too. We thought we'd talk to him about what he's been getting up to with his new Raspberry Pi 5 →

HackSpace Morning Jeff! You've done loads of Pi projects over the last couple of years. What do you still have running at the moment?

Jeff Geerling Things that are practical, really. I have Pis running all of my home network stuff, all the home lab-type things; I have a Pi running internet monitoring, so I can keep my ISP honest. I have a Pi running my VPN.

I have no cloud services for my light bulbs, or for my HVAC or anything. It's all running through Pis. I don't have any cloud account tie-in. If the internet goes down here, the only thing is I can't access it remotely, but everything's still running. I love the privacy aspects of it. And I love the ability to learn new things through it – industrial automations, and controls, and APIs with these different systems.

I think one of the things that fascinates me the most is all these different IOT devices. Before I knew much about Pis and microcontrollers, I always thought IoT was magic. And then you open one of these up, and you just see a little microcontroller inside. It might just be an ESP board, or a Pico, or something else. But I know that I could hack it, and that opens up a new world. You could build one of those on your own to just about the same quality; you can't get injection-moulded plastic, but everything else you can do on your own, and 3D printing has made it so that enclosures can be so much nicer. It's just cool to see. It's like taking down the wall of magic that you thought existed between really cool products and your own abilities.

The projects that I do are typically rehashes of what other people have done; I might make the documentation a little better or I might package it up a little nicer. But really, I would say that I'm beginner-level electronics and beginner-level microcontrollers. I just have the ability to relate what I'm doing a little better than a lot of people who are experts at it.

A year or two ago, I talked to [Linux YouTuber] NetworkChuck about recording a podcast episode. He does a lot of stuff with Raspberry Pis, too, with red hat hacking and black hat stuff. And, you know, fun things. So we recorded the podcast. And it never went up because one of the microphones was messed up, and it sounded terrible. I didn't hear from him for a while, and then on 01 March this year, he's just like, 'Hey, Jeff, do you want to work with me on a Mr. Beast project? At like, 9 am, right now?' I had no travel planned and no deadlines, so I said yes.

And that was terrible, but also awesome. So many people who make things, especially if you make it for production use, want to make sure it actually works. So there's a deadline attached to it. There are problems; there are always challenges, things you didn't even think about. And that's what happened on the Mr. Beast project. But that's also kind of the addiction that drives us forward. Because while I would say I don't want to ever do that again, I also say that was a fun experience. And getting to meet all the people there. And all of us going through that at the same time, solving all the challenges. It's a weird kind of addiction that we have, I think, building software, building hardware, and working on something so big as well.

> "Raspberry Pi devotes a lot of time to testing and to making sure manufacturing is better"

Any time we solved a problem with a five-minute fix, the sheer number of computers we were using turned that into a multiple-hour fix.

HS Why Raspberry Pi? What's so good about it?

JG Raspberry Pi devotes a lot of time to testing and to making sure manufacturing is better. And manufacturing in the UK is a pretty cool thing. How many computer companies manufacture things in their home country – there's not many, and Raspberry Pi does on a huge scale. So there's a lot of those things that some people don't assign value to – not just the bits and the little circuits on the board and all that.

Open source is another balance that Raspberry Pi has to offer. There's the Broadcom chip [which isn't open-source] versus you're building the operating system based on Debian, which is like one of the most open-source Linux distributions. And it's a weird balance. In the hardware world, it seems like there's a lot less open source, because if you put out an open-source design and someone else makes it, all of a sudden you have zero revenue. Versus software, there's sales and support and services, and there's a lot more revenue opportunities that can't just be immediately consumed by another company. Although, recently we've seen Amazon doing that sometimes, and leading to licences like the BSL, which is not really open-source.

HS I've heard of the BSD licence, but not BSL.

JG Most open-source licences don't care if you're a government or a spy or a bad person or a good person – it's free. It's unencumbered. But the BSL imposes a restriction: if you're a big business, you can't use it. Philosophically I can understand that stance, but don't call it ›





Above ↑
Jeff exploring the Sony factory where Raspberry Pis are baked

Left ←
Jeff describes himself as an electronics beginner; something tells us he's being modest



Below
Testing hardware by dropping a bowling ball on to it. Always wear eye protection!

open source: call it 'source available' or, you know, 'maker friendly'. But on the flip side, there's so many companies that start as a maker, somebody who designs a little thing, that thing becomes popular, and all of a sudden you're a big business and your licence is void, just because you hit a revenue number. And it doesn't matter if it's 50 bucks or $5 billion, it's a restriction and it's a philosophical thing that's not compatible with my understanding of the term 'open source'.

I think Raspberry Pi offers a little bit of a different take on it. Early on, a lot of the hardware was super locked down and there weren't datasheets. And one by one, they start making things better at release. So, you know, my hope is that someday we could get more open firmware and stuff for the Broadcom chips. I don't know if that'll ever happen. But it would be cool. Unlike many companies, where they start out open source and then start closing things, Raspberry Pi has always gotten better, even if it's not perfect.

HS Hopefully people will like the new power button.

JG For me, the power button on the Pi 5 is the most appreciated new feature. It's just so handy. Like, if I'm running a Raspberry Pi headless, with no keyboard, I can just do a quick shut down. I mean, I put power buttons on GPIO before, but it's cumbersome. This is nice to have it on the board.

HS And have you looked into the dual camera possibilities?

JG So machine vision has gotten really interesting in the past year, to the point where there are open-source tools to do so many things that used to require tons of expensive I/O. You can do stereo camera depth vision with all open-source tools, and [I'm] hoping to use Google Coral TPU to do some of this stuff.

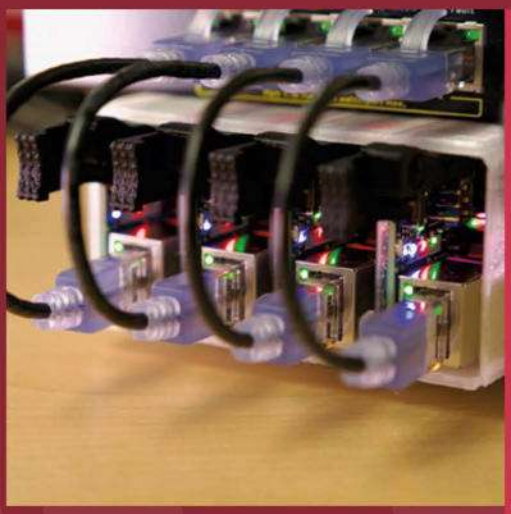
Again, it breaks down that barrier between you and the magic. You pick up an iPhone 15 and it can do depth mapping with whatever that sensor is. And you're like, man, I'll never be able to do that.

And then you get a Pi 5 with Wi-Fi and two cameras, and you're like, oh, I can plug these open-source libraries together. And we get this – it might not be as high resolution, and it might not be 60 frames per second, but you could do depth mapping and 3D stuff with a little board, and everything's under 200 bucks, all in, including power supplies and everything. That's a big difference.

When I was a kid, you could go to RadioShack and buy an electronics hobbyist kit. And that's how I learned the basics of resistors and all that. You put them together with little spring contacts, and it was really fun. And you could make buzzers and radios and all kinds of stuff – they don't really exist anymore. All those hobbies that were big when I was a kid, a lot of them are superseded just by how amazing the technology we have today is. But the Raspberry Pi brings that back.

> "For me, the power button on the Pi 5 is the most appreciated new feature. It's just so handy"

What's funny is that the Raspberry Pi is actually better at IoT stuff than most IoT companies. Mostly because of the open source philosophy. If we all work together, all these individuals around the world can write plug-ins and do stuff where you can put home assistant at your house. And you can interact with any light bulb, any doorbell, any camera,



Above ↑
Anyone can plug wires in; it takes skill to get hardware working together

any washing-machine controller. Whereas if you get Philips, you get the Philips stuff and if you get Sony, you get the Sony stuff.

It's like the tool world: I use DEWALT mostly, because the first tool I bought was a DEWALT. So you gotta use the DEWALT battery, and then you're like, well, I could buy this cheaper tool, it's better. But then it won't work with my battery. So now I have something like 25 DEWALT tools. I think that's dumb. It's ridiculous. And that's how IoT people want to be: you have the smart hub from this company, and you buy all their stuff. But we've already seen companies fail, and then all of your smart stuff becomes dumb. And it's worse than dumb because you can't even turn on a light bulb, you know – their server goes down and your lights turn off.

This is an area where instead of just seeing behind the curtain how it's done, you can do it better. That's one thing that I love about the Raspberry Pi ecosystem and the open-source ecosystem together. →

HS A slightly less practical, though no less awesome build of yours is the Pi cluster. Just one question: why?

JG For me, it's fun. You get something out of it when you're doing all this work on a cluster. And the first time you see all the nodes come up together, that's kind of magic. And then also seeing the fact that, like, you can actually program things to scale. It's useful for a job, especially if you're gonna build software for the web.

It was a springboard for learning. And for a lot of us, we love that. We love hacking with things, and a Pi cluster is just a fun thing to do it. It also has the side benefit of being able to develop skills that might be incredibly useful. There are still a lot of companies that will pay good money for Kubernetes developers, and a lot of people have learned Kubernetes on Raspberry Pi. The shortage really built into that. And a lot of people started using small PCs that they got on eBay, which is perfectly fine. I think that's awesome. The hard thing was, like, the reason the Pi was great was because it's a $35 computer – now a $60 computer – but you can still get the Pi 4, and hopefully there'll be more available now that Pi 5 has been announced. But it was a quick way to get into that whole ecosystem of learning. And, you know, it's not necessary at all for a home lab. But it's fun.

And it translated into me getting a better job and doing consulting, which also helped me with that opportunity that I got in the pandemic: I was consulting and making extra money from doing all the work I was doing on Pis, learning clustering, because the clustering skills got me a better-paying job. And it gave me the opportunity to save up some money over the next year. I went to full-time YouTube in 2021, so maybe the Pi cluster was the reason I got into it.

I did my first Pi cluster in 2016. And I had a cluster running from 2016 to 2020 24/7 at my house, and I think had something like two hours of downtime in that time period.

HS You said that you're not a hardware guy. Does that mean that you have a background in software?

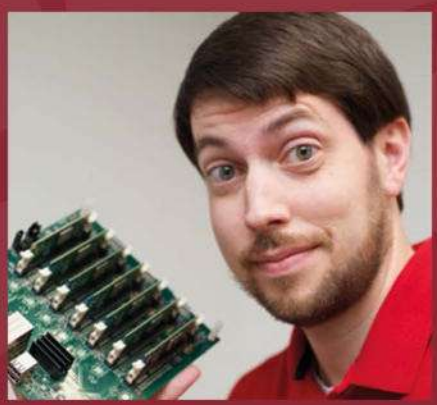
JG I started off doing web design. The first ever project I did was helping a radio station. They just transitioned to computers instead of CDs for their music playback, and in their system it had a file that would write to with the current song. My dad had the idea of taking this data and putting the song name on the website, so people could be listening, go to the website, on their laptop, or in an office, back then people did not have smartphones. It was ahead of its time. So I built a little interface that had, like, a car radio with the song title in it in a little, I think it was in Courier font or something, because there were only ten or so fonts that you could use back then. It worked for five or six years, until they got a real website. But, that was my first-ever project.

> "I did my first Pi cluster in 2016. And I had a cluster running from 2016 to 2020 24/7 at my house"

I learned electronics from my dad, who's a radio engineer. And I, you know, I soldered together an FM radio. I still have the voltmeter that I soldered together when I was, like, eight years old or something. It still works, and I keep using it, even though it's not as accurate as it could be. And I'm excited to do more of that since the Pico came out. I've done some more projects: I built a garage door sensor thing, and I'm working on some software for the Pimoroni Galactic Unicorn to write some stuff on the wall. And now I have a new office that I'm gonna go into. So there's a lot of opportunities to build stuff for that.

HS That's a majestic piece of kit.

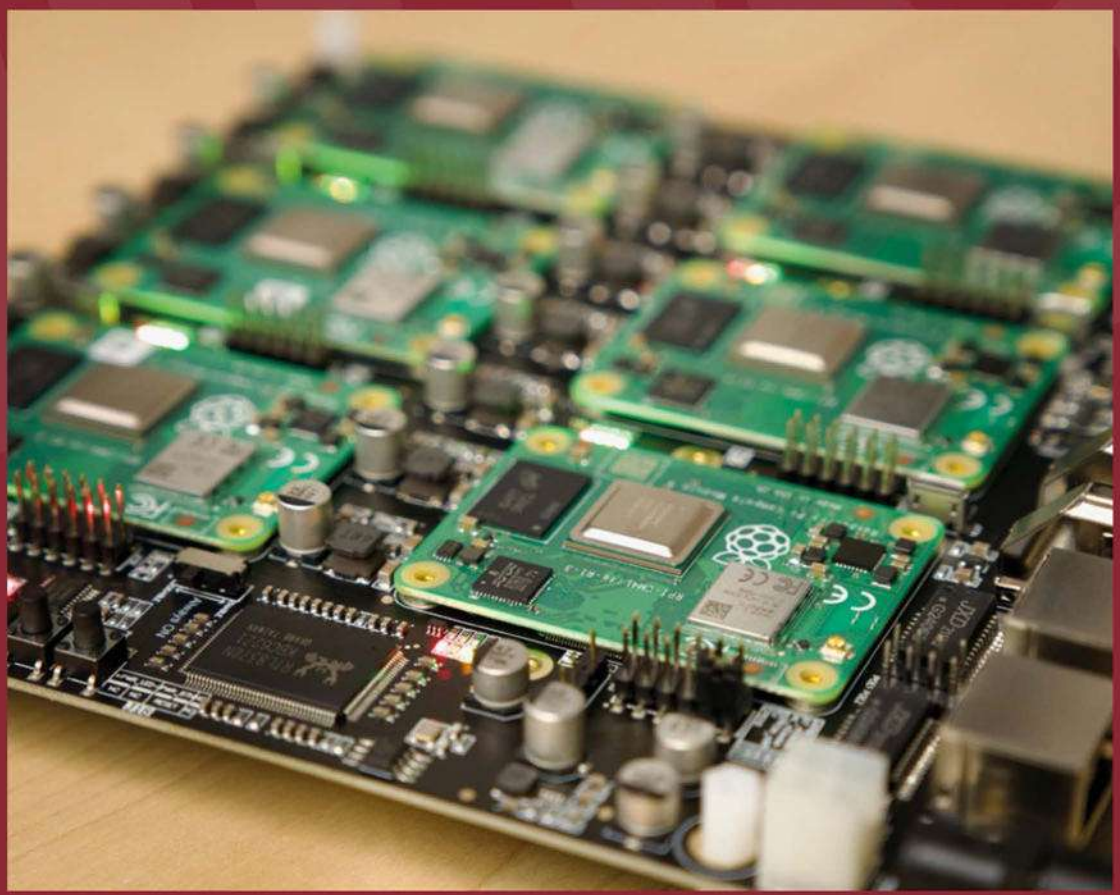
JG Yeah. And they put in little things like a button. You know, it's nice to have a button here – now you can do, like, modes for the display. And you can even make a little game on the LEDs matrix. That's the other thing that I love about the Pi ecosystem: Raspberry Pi works directly with some of these companies to make things better for everyone. And these companies work back with them. It's a relationship that has been built up over the course of the past decade. It's bearing more fruit with the Pico, and I think the Pi 5 will drive some of that, too. I'm really excited to see what people come up with with HATs for the Pi 5 with PCI Express. If anybody comes out with a dual 2.5 gigabit network cat, I'm gonna buy that thing the second I see it.



Above
Jeff's fascinated by factories. And he loves open source software and hardware

Above ↑
The Petabyte Pi: a Raspberry Pi with 1024 terabytes of storage

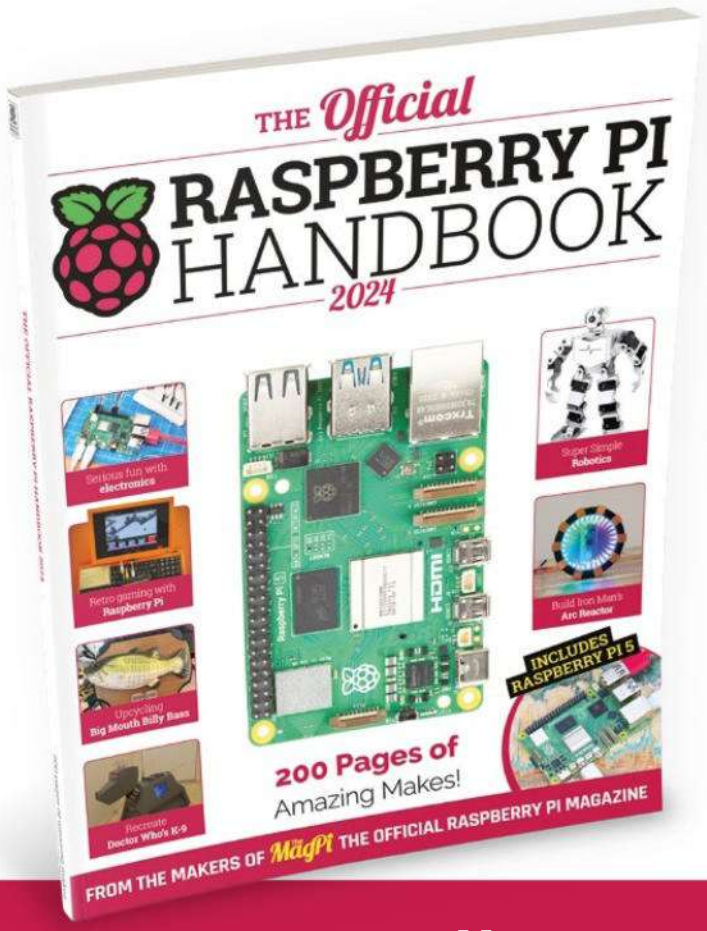
Right →
Why build a Raspberry Pi cluster? Why not!



THE Official RASPBERRY PI HANDBOOK 2024

200 PAGES OF RASPBERRY PI

- QuickStart guide to setting up your Raspberry Pi computer
- Updated with Raspberry Pi Pico and all the latest kit
- The very best projects built by your Raspberry Pi community
- Discover incredible kit and tutorials for your projects







Buy online: magpi.cc/store

HackSpace
TECHNOLOGY IN YOUR HANDS

FORGE

HACK | MAKE | BUILD | CREATE

Improve your skills, learn something new, or just have fun tinkering – we hope you enjoy these hand-picked projects

PG 58

PICO QUIZ

Get interrogated by your microcontroller



PG 64

LIGHT BURN

Upgrade your laser cutter

PG 53

SCHOOL OF MAKING

Start your journey to craftsmanship with these essential skills

54 Synthio



PG 70

AUTOMATIC APPLAUSE

Designing a linkage

PG 74

POWER MONITOR

Make the most of cheap electricity

Making music on Pico

Use CircuitPython to create complex sounds



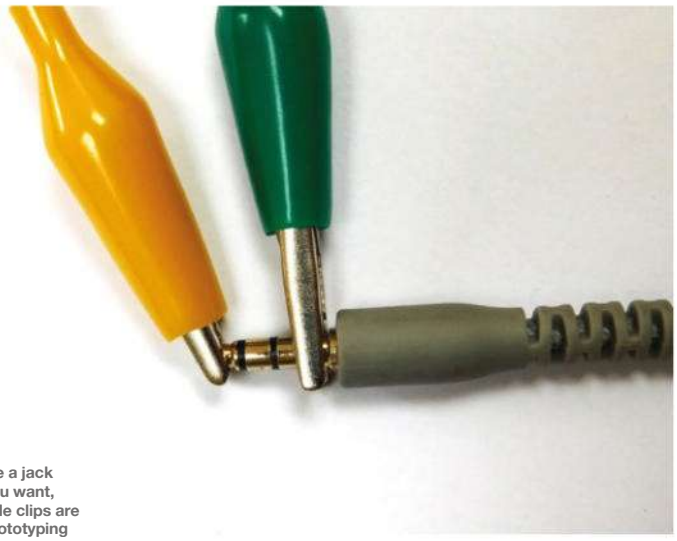
Ben Everard

Ben's house is slowly being taken over by 3D printers. He plans to solve this by printing an extension, once he gets enough printers.

Most microcontroller programming environments have a way of making beeps, and even if there's not an in-built method, it's a pretty simple thing to do – just flick a pin on and off quickly (but not too quickly) and you have an audio signal that you can send to a speaker, headphones, or other audio devices.

CircuitPython, however, has just gained a complex audio synthesis module called synthio. This lets you not just create beeps and boops, but control these sounds in a variety of ways. Let's take a look.

You'll need a board that supports synthio. RP2040 boards, such as Pico and Pico W, are a great option for testing things out and having a play using headphones; however, a board with dedicated audio hardware (such as the Adafruit RP2040 Prop-Maker) will make it easier to connect a speaker.



Above
You can use a jack socket if you want, but crocodile clips are great for prototyping

Before we dive into the code, let's take a look at the hardware. If you're using a board without dedicated audio hardware (such as a Pico), you just need to connect a pair of headphones between one of the GPIO pins and ground. The easiest way to do this is to clip crocodile clips onto the headphone jack.

You can probably get away without adding a resistor in there as well, but it's a bit safer to put a 220Ω resistor between the GPIO pin and the headphones. This stops the current from getting too high (especially if you are using crocodile clips that can accidentally touch).

Ideally, you should have a capacitor between the GPIO pin and ground. This is because the signal coming from the GPIO pin isn't an audio signal, but a high-frequency PWM signal that, when averaged out, becomes the audio signal. The capacitor creates a low-pass filter that does this averaging out. We find that this isn't generally necessary because the rest of the circuit tends to act as a low-pass filter, so it works fine without it.

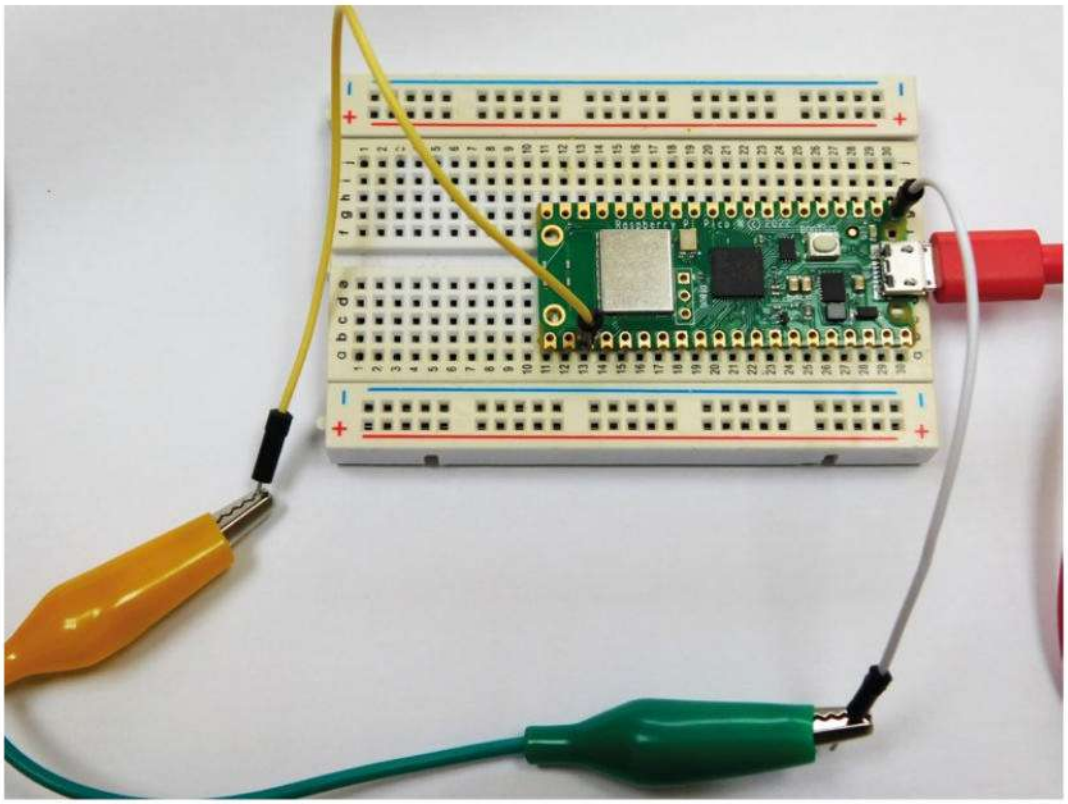
We used GPIO0, but you can adjust this to any you want.

Let's take a look at a really simple example:

```
import time
import board
import digitalio
import synthio
import audiopwmio
import audiomixer

audio = audiopwmio.PWMAudioOut(board.GP0)
mixer = audiomixer.Mixer(channel_count=1, sample_
rate=22050, buffer_size=2048)
synth = synthio.Synthesizer(sample_rate=22050)

audio.play(mixer)
```



Left
You can get away without a resistor, but it's a bit safer to add one in line with the GPIO pin

```
mixer.voice[0].play(synth)
mixer.voice[0].level = 0.1

while True:
    synth.press(65)
    time.sleep(0.5)
    synth.release(65)
    time.sleep(1)
```

Here, we use three objects. There's a **PWMAudioOut** which handles the output. There are other output options if you have additional hardware, for example, if you want to use an I2S DAC. There's a **mixer**. Technically, we could get away without this, in which case, the audio would play at full volume, but this can be a bit loud on headphones, so we prefer to always use a mixer to enable us to keep control over how loud we go. In future examples, you'll also see how a mixer can be used to combine sounds. Finally, there's a **synth** which is used to generate the sound.

In this case, we just loop through, pressing and releasing note 65. The notes are given numbers in the MIDI system, and 65 is F#.

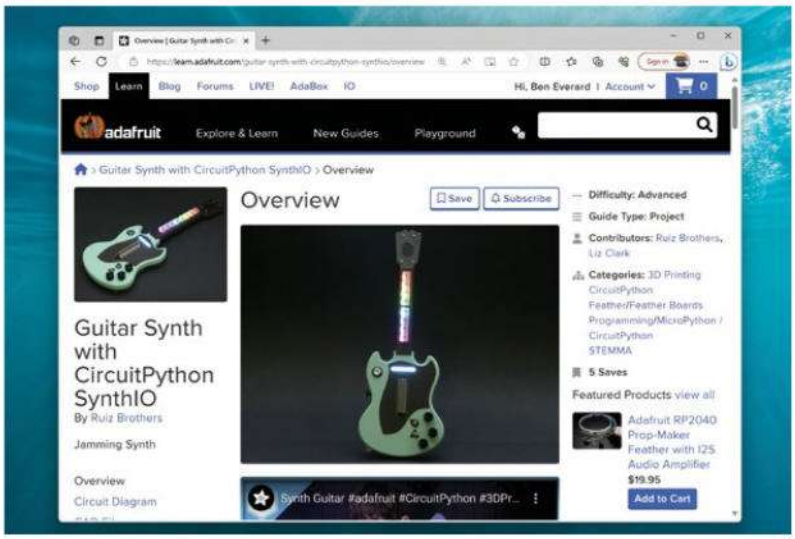
When we press it, it goes on full; when we release it, it goes off instantly. This is OK, but it sounds a bit harsh. Wouldn't it be better if we could fade the sound in and out? Synthio has a solution to this – it's called envelopes:

```
import time
import board
import digitalio
import synthio
import audiopwmio
import audiomixer

audio = audiopwmio.PWMAudioOut(board.GP0)
mixer = audiomixer.Mixer(channel_count=1, sample_
rate=22050, buffer_size=2048)
synth = synthio.Synthesizer(sample_rate=22050)
audio.play(mixer)
mixer.voice[0].play(synth)
mixer.voice[0].level = 0.1

amp_env = synthio.Envelope(
                                attack_time=0.05,
                                sustain_
level=0.2,
                                release_time=0.5
)

while True:
    synth.envelope = amp_env
    synth.press(46)
    time.sleep(1.25)
    synth.release(46)
    time.sleep(1.25)
```



Above There's an example guitar created by the Ruiz Brothers at hsmag.cc/synthioguitar

Envelopes are a way of controlling the volume over time using the parameters: **attack_time**, **attack_level**, **sustain_time**, **sustain_level**, **release_time**, **decay_time**.

At first, the volume will ramp up to **attack_level** over the period of **attack_time**. Then it will drop down to **sustain_level** volume over **decay_time**. It will then stay at the same volume for **sustain_time**, and finally drop down to zero volume over **decay_time**. All times are in seconds, and volumes are between 0 and 1.

Now we can play notes that sound a bit more natural, let's take a look at combining them. The following code uses two **synth** objects to play notes at different times:

```
import time
import board
import digitalio
import synthio
import audiopwmio
import audiomixer

audio = audiopwmio.PWMAudioOut(board.GP0)
mixer = audiomixer.Mixer(channel_count=1, sample_
rate=22050, buffer_size=2048)
synth1 = synthio.Synthesizer(sample_rate=22050)
synth2 = synthio.Synthesizer(sample_rate=22050)

audio.play(mixer)
mixer.voice[0].play(synth1)
mixer.voice[0].level = 0.1
mixer.voice[1].play(synth2)
mixer.voice[1].level = 0.05

amp_env = synthio.Envelope(
                    attack_time=0.05,
                    sustain_level=0.3,
                    release_time=0.3
)

amp_env2 = synthio.Envelope(
                     attack_time=0.05,
                     sustain_level=0.5,
                     release_time=0.5
)

beat = 0
synth1.envelope = amp_env
while True:
    if beat % 2 == 0:
        synth1.press(60)
    if beat % 2 == 1:
        synth1.release(60)
    if beat % 4 == 0:
        synth2.press(46)
    if beat % 4 == 2:
        synth2.release(46)
    time.sleep(1)
    beat += 1
```

We can't use a simple repeating loop here because that wouldn't let us play notes at different rates, so we've created a beat loop, then we can detect which beats we want to start and stop different notes at.

ON ANOTHER NOTE

So far, we've set the volume structure of the note, but we've still been playing the same sound – a square wave beep. We can create far more complex

> **Now we can play notes that sound a bit more** natural, let's take a look at combining them

sounds than this. Synthio uses a wavetable function to create the audio signal. This basically means that you can create an array of volumes that define the shape of the audio signal for a note. This could go abruptly from 0 to 1 as in a square way; it could go up linearly like a triangular wave; it could have a curve like a sine wave, or just about anything else. Each wave shape produces a very different sound.

```
import time
```

```
import board
import digitalio
import synthio
import audiopwmio
import audiomixer
import ulab.numpy as np

audio = audiopwmio.PWMAudioOut(board.GP0)
mixer = audiomixer.Mixer(channel_count=1, sample_
rate=22050, buffer_size=2048)
synth1 = synthio.Synthesizer(sample_rate=22050)
synth2 = synthio.Synthesizer(sample_rate=22050)

audio.play(mixer)
mixer.voice[0].play(synth1)
mixer.voice[0].level = 0.2
mixer.voice[1].play(synth2)
mixer.voice[1].level = 0.05

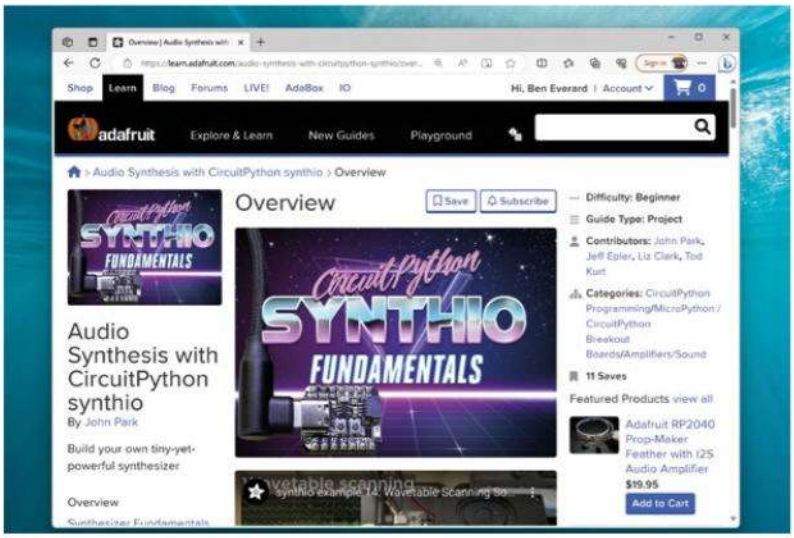
amp_env = synthio.Envelope(
                          attack_time=0.05,
                          attack_level = 1,
                          decay_time = 0.1,
                          sustain_level=0.3,
                          release_time=0.3
)

amp_env2 = synthio.Envelope(
                           attack_time=0.05,
                           attack_level = 1,
                           decay_time = 0.1,
                           sustain_level=0.1,
                           release_time=0.1
)

SAMPLE_SIZE = 512
SAMPLE_VOLUME = 32000  # 0-32767
half_period = SAMPLE_SIZE // 2
wave_sine = np.array(np.sin(np.linspace(0, 2*np.
pi, SAMPLE_SIZE, endpoint=False)) * SAMPLE_VOLUME,
                    dtype=np.int16)

note1 = synthio.Note(synthio.midi_to_hz(60),
waveform=wave_sine, amplitude=1)

beat = 0
synth1.envelope = amp_env
while True:
    if beat % 2 == 0:
        synth1.press(note1)
    if beat % 2 == 1:
        synth1.release(note1)
```



Above
How far do you want to disappear down the synth rabbit hole?

```
    if beat % 4 == 0:
        synth2.press(46)
    if beat % 4 == 2:
        synth2.release(46)
    time.sleep(1)
    beat += 1
```

We've only really scratched the surface of what you can do with synthio here. It's a hugely powerful system for creating music using microcontrollers. There's a more in-depth introduction online at **hsmag.cc/synthiofundamentals**.

SYNTH TERMINOLOGY

Synthio uses similar terminology to many other synthesizers. You'll come across envelopes and mixers similar to those we've used in this article in many other bits of software and hardware. Here are a few other things that synthio provides that you will also find elsewhere:

- Low-frequency oscillator (usually known as an LFO) is a special type of oscillator that unsurprisingly runs slowly, or at least slower than typical audio frequencies. The purpose of this isn't to generate audio sounds directly but to influence other audio. For example, you can use it to bend a note in a vibrato-like effect.
- Filters block some audio signals while allowing others to pass through. The type of filter defines what frequencies are allowed to pass through. Synthio Synthesizer objects have the following methods to create filters: high_pass_filter, low_pass_filter, and band_pass_filter. Once they are created, they can be attached to notes or synthesizer objects.

Make a Pico LCD true or false quiz game

An LCD display can display messages from Raspberry Pi Pico. In this project, the display is used as part of a quiz game



MAKER

Stewart Watkiss

Also known as Penguin Tutor. Maker and YouTuber that loves all things Raspberry Pi and Pico. Author of *Learn Electronics with Raspberry Pi*.

penguintutor.com
twitter.com/stewartwatkiss

In this tutorial, we will create an interactive true or false quiz game using a Raspberry Pi Pico and an LCD display. Whilst doing so, you will learn about some of the pitfalls when connecting to devices running at different voltages. You'll discover ways to increase voltage output using a simple buffer, and how to make a bidirectional level shifter. The level shifter is then used to convert between 3.3V for the GPIO ports on a Pico to 5V used by the LCD display. The game is programmed in Python, with a text file for the questions. The game can be installed inside an enclosure for a complete game.

You'll Need

- LCD display with PCF8574T **magpi.cc/ihaospacelcd**
- I2C safe level-converter **magpi.cc/BSS138**
- 3 × 16 mm button switches **magpi.cc/16mmbutton**

01 LCD character display

This project is based around an LCD display. Our display has 16 characters across two lines and is often referenced as a '1602'. These usually contain an HD44780, or equivalent, driver chip that displays the appropriate pixels that make up the characters.

One downside of the display is that the driver chip needs at least six data connections. This uses up GPIO ports, as well as needing lots of wires to the LCD display. A common solution is to have a 'backpack' fitted to the rear of the LCD display using a port expander. The example used here is a PCF8574T 8-bit port expander.

02 Designed for 5 V

The port expanders are available on a PCB backpack pre-soldered onto the back of the LCD PCB. This saves you from having to create your own circuit, but it does come with an issue. These circuits are normally designed for 5V, whereas a Pico uses 3.3V for the GPIO ports.

Connecting a 5V signal to a Pico GPIO port could cause permanent damage to the latter, so this tutorial looks at some of the possible solutions to interfacing between devices designed for different voltages.

03 Move pull-up to 3.3 V

If the 5V device did not have a pull-up resistor, the I2C bus could work with pull-ups to the 3.3V supply instead. This is shown in **Figure 2**. The crossed-out resistors are the pull-ups inside the LCD I2C backpack and the two pull-up resistors on the left are connected to the 3.3V output on a Pico. Unfortunately, this involves de-soldering surface-mount devices, which can be difficult.

04 Unidirectional level shifter

A simple form of level shifter can be used when controlling 5V devices from a 3.3V

microcontroller or computer. This is often used for controlling NeoPixels from a Pico or a Raspberry Pi. In its simplest form, this is a MOSFET with two resistors (as shown in **Figure 3**, overleaf). The gate resistor RG (typically 470Ω) reduces the in-rush current, and RL is a pull-up resistor (typically 2.2kΩ to 10kΩ). With no input, the pull-up resistor sets the output high. When a 3.3V input is provided, the MOSFET turns on pulling the output low. This results in an inverted signal.

The code can be configured to invert the output, or you could add an additional MOSFET to invert it a second time. A two-stage, non-inverting buffer is shown in **Figure 4**.

05 Bidirectional level shifter

The LCD is controlled from your Pico, so you may expect the signal would only need to go in one direction. However, due to the use of I2C protocol, signals need to pass in both directions. We need a bidirectional level shifter. These can be made using individual MOSFETS, but using a premade level shifter from Adafruit or SparkFun is more convenient. An example is the Adafruit bidirectional level shifter, which has four level shifters on a convenient PCB. This is shown in **Figure 5**.

The level shifter has just one MOSFET for each channel. This is in an unusual configuration. The circuit can be thought of as two sides, with the left side being for the low voltage and the right for the higher voltage. The MOSFET joins the two together. The schematic diagram is shown in **Figure 6**.

06 How the level shifter works

If both the low-voltage and high-voltage signals are high, then the MOSFET is off and the signal is high at both sides. If the low-voltage signal (left) drops low, then the MOSFET is in the forward direction and the voltage at the gate will turn the MOSFET on. This will provide a path to ground and so the high-voltage signal (right) will be pulled low. If the high-voltage signal (right) goes low, due to an internal characteristic of the MOSFET a small current is able to flow in the reverse direction. As this happens, the voltage of the source pin dips, causing the MOSFET to turn on. This pulls the voltage down on the low-voltage signal as well.

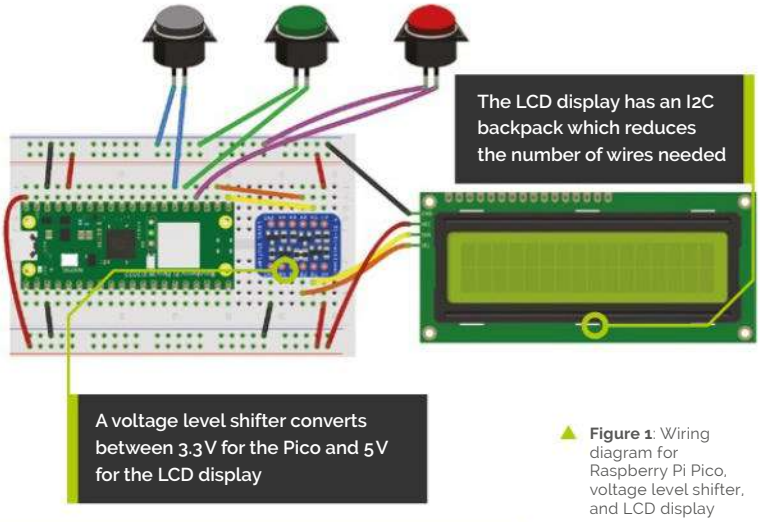


Figure 1: Wiring diagram for Raspberry Pi Pico, voltage level shifter, and LCD display

07 LCD circuit

The level shifter can be inserted onto the breadboard and connected between your Pico and LCD display. Then it's just a case of adding three buttons for Start, True, and False. These are shown in **Figure 1**.

The top power rail is used for 3.3V taken from your Pico's 3.3V output, and the bottom power rail is 5V taken from the VBUS supply from the USB port.

The buttons used are 16mm push-to-make switches, similar to arcade buttons, but smaller. You can use other push-to-make switches if you prefer.

08 Download the LCD library

The libraries that support the LCD display with backpack are available from GitHub (**magpi.cc/rpipicoi2clcd**). Upload the files

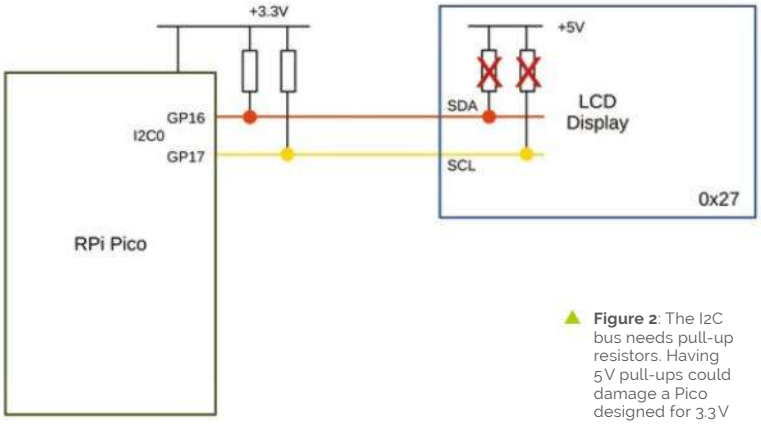


Figure 2: The I2C bus needs pull-up resistors. Having 5V pull-ups could damage a Pico designed for 3.3V

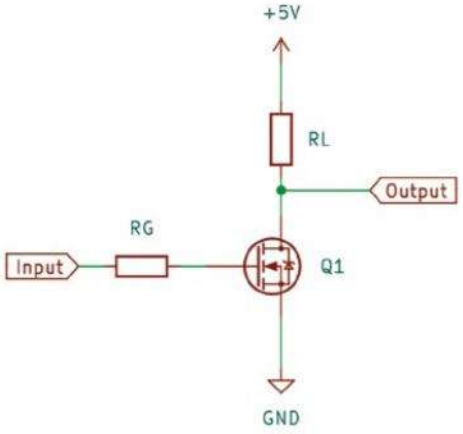


Figure 3: A simple MOSFET level shifter. The output is the opposite of the input but higher voltage

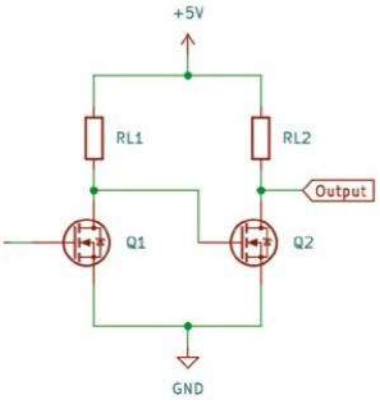


Figure 4: A second MOSFET can be used to create a non-inverting buffer

lcd_api.py and **pico_i2c_lcd.py** to your Pico. You can see a demo using **pico_i2c_lcd_test.py**. This can be useful for checking your wiring is correct, but you will need to change the pins used for SDA (GPIO 16) and SCL (GPIO 17).

09 Coding the game

The game code (**quizgame.py**, overleaf). starts by setting up the three `button` objects, along with `i2c` and `lcd`. It then reads the file **quizfile.txt**, which contains the questions.

Then it enters a loop which ensures that the game can be played over again.

Within the first few lines of the loop, you can see that it first clears the display, puts a string which starts on the top line, moves to the start of the second line, and then puts another string to that line.

10 Handling button presses

The button presses are handled by having a `while` loop which runs until an appropriate button is pressed. In the case of the Start button, it just looks for that one button, but when waiting for a true and false, it needs to check both the `true_button` and `false_button` to see if either is pressed. It keeps track of the score and then displays the score at the end, pausing for five seconds before restarting the game.

11 The quiz file

The questions are stored in the file **quizfile.txt**. This has one line per question. Each line should have three entries separated by a semicolon. The first entry is the top line to display, the second is the second line, and the final entry is a letter T or F to indicate whether the correct answer is `True` or `False`.

> "The code to create the game is included"

The file is opened using the `with` statement. Using with means that the file will be automatically closed after the program has finished reading in the entries. The `readlines` method is used to read all the entries into a list.

To separate the text to display from the answers, the `split` method is used. You may notice that it also uses the `strip` method to ignore any whitespaces, such as spaces before the newline character.

The quiz file is created separately and must be uploaded to Pico.

12 Improving the game

The game can be placed in an enclosure to make a complete game. You could start with a standard enclosure and cut holes for the display and buttons, or if you have a 3D printer you can

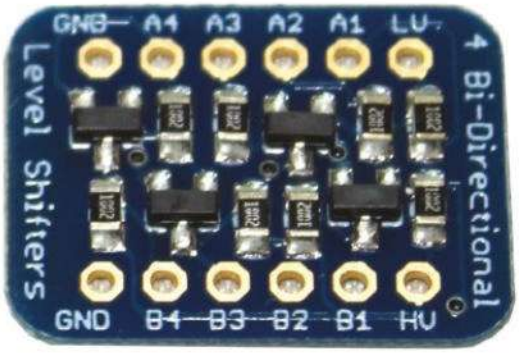


Figure 5: A bidirectional level shifter is available on small PCBs. These can have headers fitted to be used on a breadboard

THE MAGPI



This tutorial is from in The MagPi, the official Raspberry Pi magazine. Each issue includes a huge variety of projects, tutorials, tips and tricks to help you get the most out of your Raspberry Pi. Find out more at **magpi.cc**

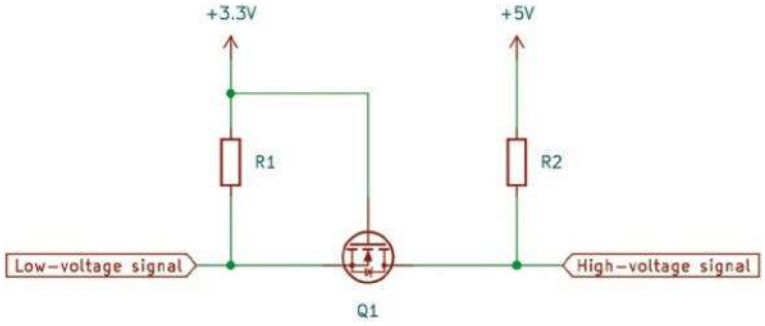


Figure 6: The bidirectional level shifter uses two MOSFETs. This works well for signals with pull-up resistors like I2C

download an example from the Penguin Tutor website (**magpi.cc/trueorfalse**). One improvement would be to add some error checking. Without error checking, if there is an invalid entry in the quiz file, the program may crash.

Another possible improvement would be to provide a way to add multiple quizzes rather than just limiting them to a single quiz.



Figure 7: The game can be placed inside an enclosure. If you don't have a 3D printer, you can use a generic case and cut appropriate holes

quizgame.py

DOWNLOAD THE FULL CODE:
magpi.cc/picolcdquiz

> Language: **MicroPython**

```
001. import utime
002. from machine import Pin, I2C
003. from lcd_api import LcdApi
004. from pico_i2c_lcd import I2cLcd
005.
006. I2C_ADDR     = 0x27
007. I2C_NUM_ROWS = 4
008. I2C_NUM_COLS = 16
009.
010. start_button = Pin(20, Pin.IN, Pin.PULL_UP)
011. true_button = Pin(19, Pin.IN, Pin.PULL_UP)
012. false_button = Pin(18, Pin.IN, Pin.PULL_UP)
013.
014. i2c = I2C(0, sda=machine.Pin(16),
     scl=machine.Pin(17), freq=400000)
015. lcd = I2cLcd(i2c, I2C_ADDR, I2C_NUM_ROWS,
     I2C_NUM_COLS)
016.
017. lcd.hide_cursor()
018.
019. # Read questions into a list
020. with open("quizfile.txt", 'r') as file:
021.     questions = file.readlines()
022.
023. while True:
024.     lcd.clear()
025.     lcd.putstr("True or False")
026.     lcd.move_to(0,1)
027.     lcd.putstr("Press Start ...")
028.
029.     while True:
030.         if (start_button.value() == 0):
031.             break
032.     # Quiz start
033.     score = 0
034.     for question in questions:
035.         lcd.clear()
036.         # strip off any whitespace
037.         # then split the entries into line1, 2 and
     answer
038.         (text, text2, answer) =
     question.strip().split(";", 3)
039.         lcd.putstr(text)
040.         lcd.move_to(0,1)
041.         lcd.putstr(text2)
042.         while True:
043.             if (true_button.value() == 0):
044.                 if (answer == "T"):
045.                     score += 1
046.                 break
047.             if (false_button.value() == 0):
048.                 if (answer == "F"):
049.                     score += 1
050.                 break
051.     lcd.clear()
052.     lcd.putstr("Game over")
053.     lcd.move_to(0,1)
054.     lcd.putstr("Score {} of {}".format(
     score, len(questions)))
055.     utime.sleep(5)
```

RASPBERRY PI 5



Priority Boarding

We've reserved Raspberry Pi 5 boards
for HackSpace magazine subscribers

GET YOUR RASPBERRY PI 5 FIRST!

Want to get Raspberry Pi 5 sent to you as soon as it's released? With Priority Boarding, you can pre-order your Raspberry Pi 5 (4GB or 8GB) and it'll be sent out in the first week.

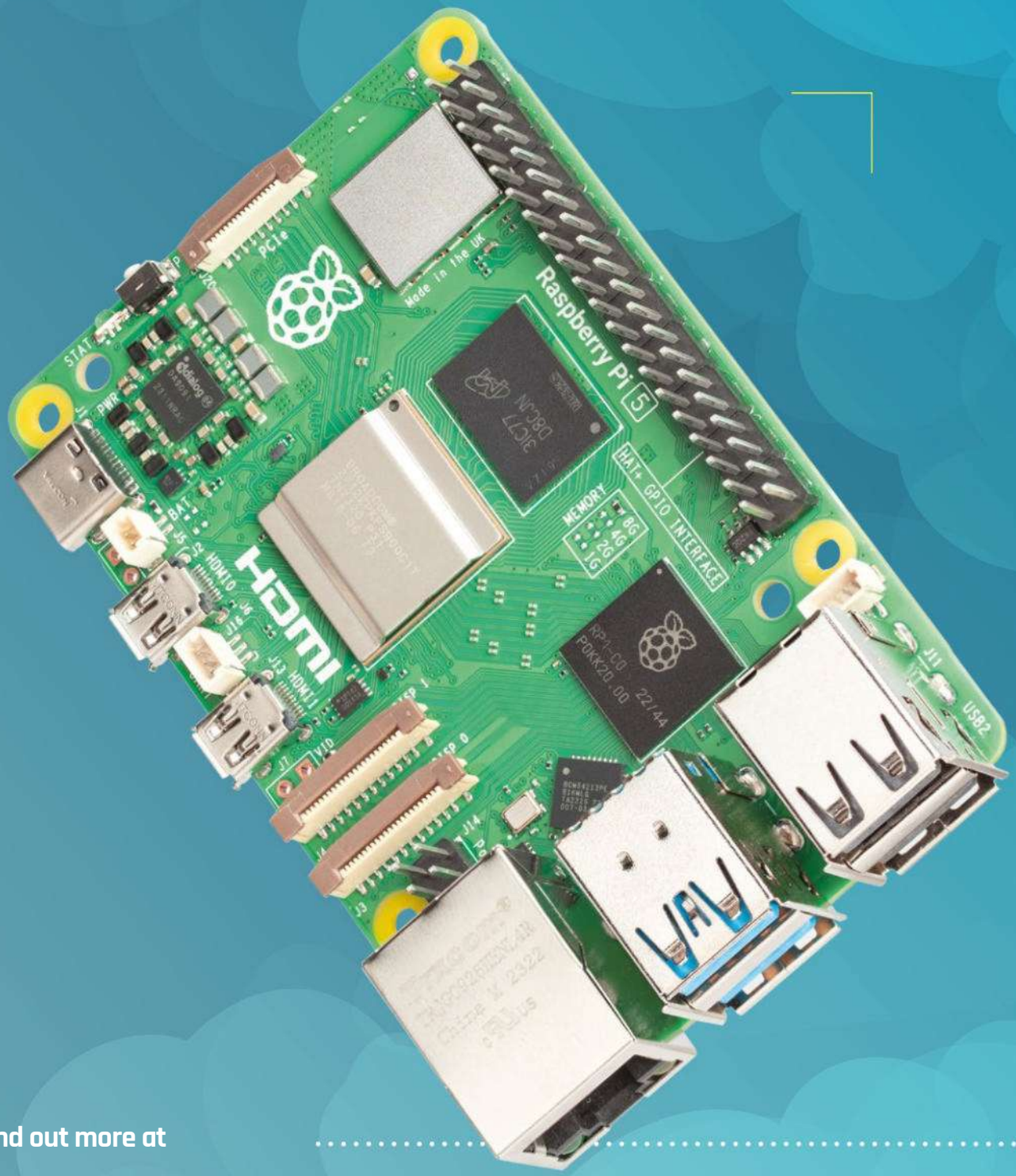
Raspberry Pi has set aside thousands of Raspberry Pi 5 computers for The MagPi and HackSpace magazine print subscribers to buy. Enough to guarantee every subscriber can get one without going out of stock.

How it works!

If you have a print subscription to HackSpace magazine or The MagPi, we will have your email on file. We'll send each of our subscribers a unique code in the next few days. Along with the code, you'll get details of how to use it through our partners (Pi Shop in USA and Canada and the Pi Hut everywhere else).

Get started

New subscribers to HackSpace mag will get a code when they sign up. So don't delay; take out a subscription today (**hsmag.cc/subscribe**). You'll be able to buy your Raspberry Pi 5 first and get a magazine packed full of incredible maker tutorials, projects, and reviews.



Find out more at

hsmag.cc/prioritybording

Terms & Conditions Priority Boarding codes will be emailed to everybody with a print subscription to The MagPi or HackSpace magazine. People who subscribe to both magazines (print only) will receive two codes. Priority boarding does not apply to people with App Store, Google Play, ZINIO, PDF contributions, or other paid-for subscriptions. Each code will entitle you to purchase 1 × Raspberry Pi 5 model (4GB or 8GB) for the standard retail price and delivery. Multiple codes need to be used individually. This is a limited offer and is subject to change or withdrawal at any time.



Controlling the K40 laser cutter with LightBurn

Make the K40 speak GRBL



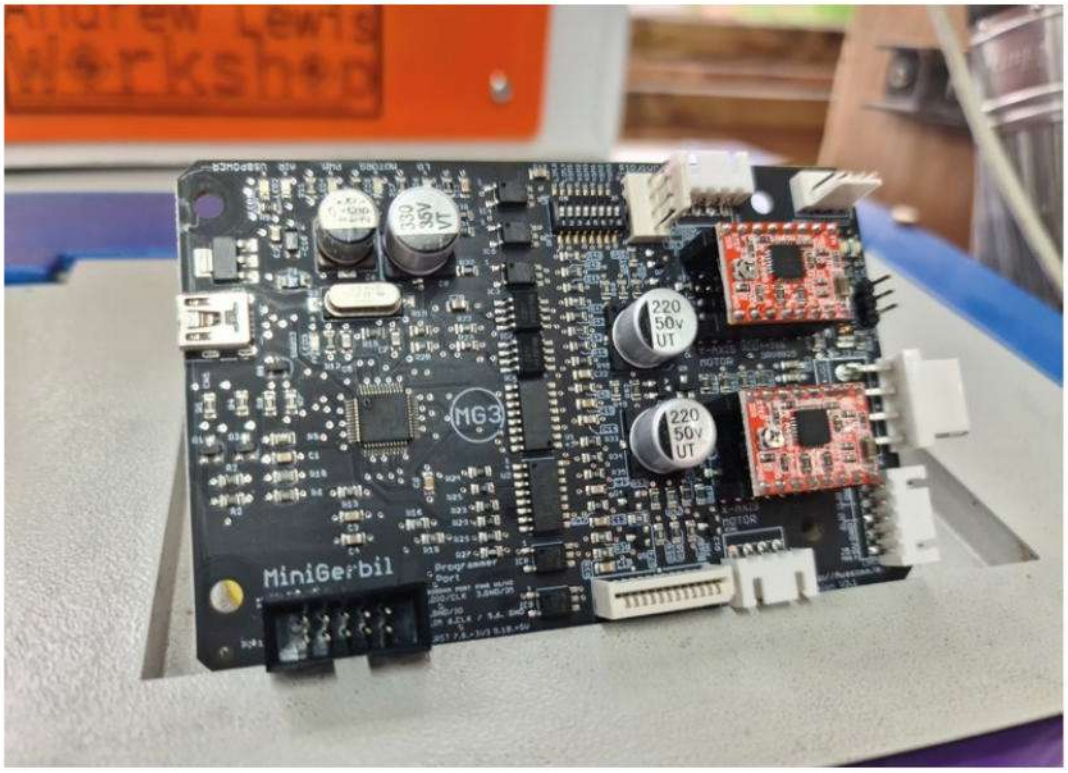
Dr Andrew Lewis

Dr Andrew Lewis is a specialist fabricator and maker, and is the owner of the Andrew Lewis Workshop.

One of the biggest complaints that people have with the K40 laser cutter is that it doesn't speak proper G-code. It speaks a sort of proprietary language that isn't really compatible with anything except for the software it ships with. A couple of open-source packages like K40 Whisperer have managed to learn to speak K40 and, while these packages are great, they don't have the features and the polished interface that a commercial product like LightBurn can offer. In this project, you'll find out how to replace the controller of your K40 so that it speaks a more conventional dialect of G-code and can interface with applications like LightBurn.

STAY SAFE

A laser cutter is a dangerous machine, and a partially disassembled laser cutter is doubly so. Unplug the machine from the mains before you open the controller enclosure, and then proceed under the assumption that the machine is actively trying to kill you. You can never be completely certain that the machine is dead. Capacitors hold power for a long time after a machine is unplugged, the wrong plug could have been pulled, or the wrong isolator switch could have been switched. Invisible laser radiation can disfigure or permanently blind you. Always wear laser safety goggles when working with the laser. Never leave the laser in a state where it could be powered up accidentally.



Left
The Mini Gerbil from Awesome.Tech has connectors that match those on the original control board, which makes upgrading exceptionally easy. Additionally, there are plenty of extra pins that facilitate other functions like automatic air assist management and pause/resume controls

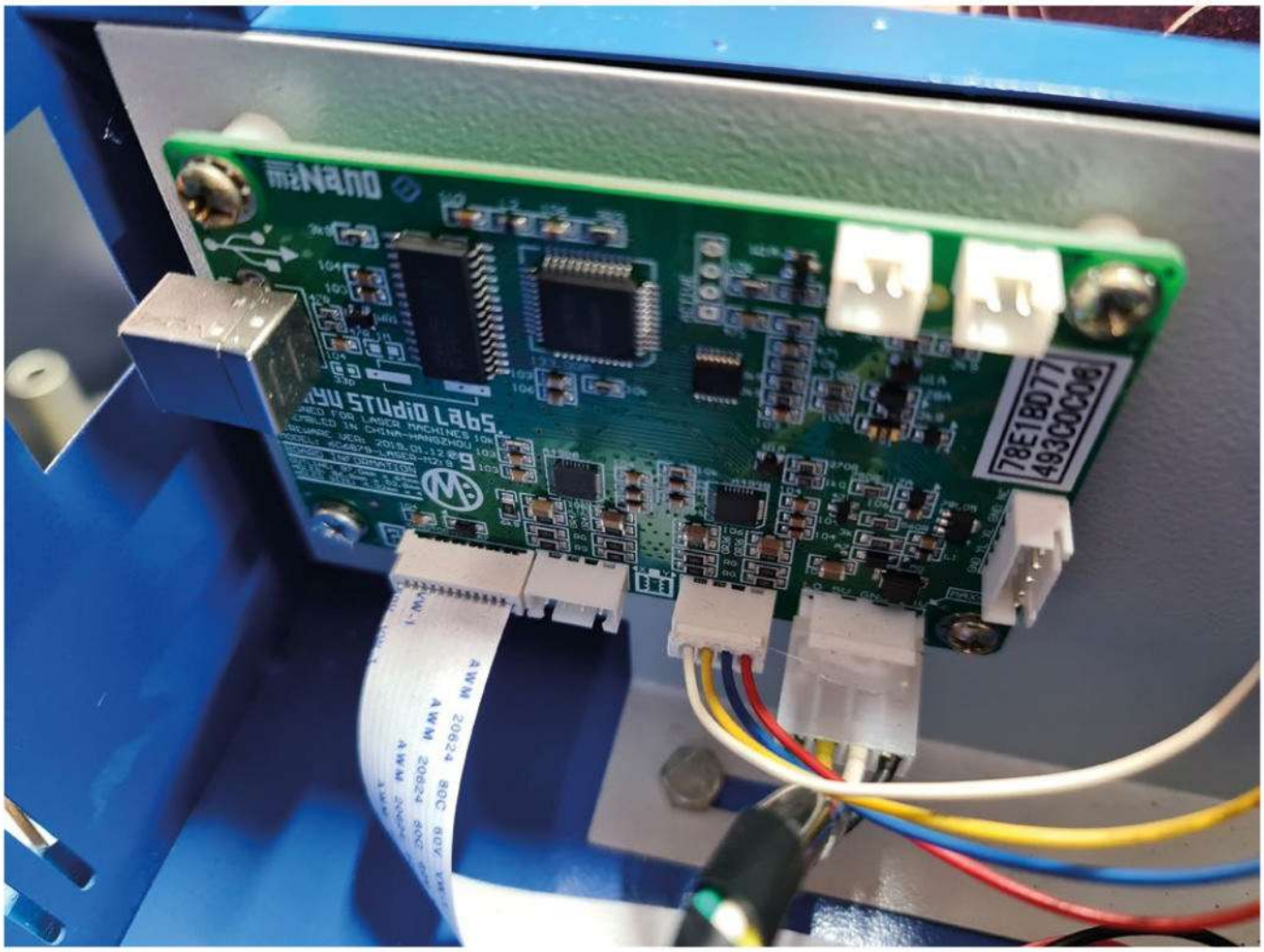
> **The stock controller is built to be the cheapest possible** functional option

NEVER TRUST A LIVE LASER

Like every other component on the K40 laser, the stock controller is built to be the cheapest possible functional option. Aside from the proprietary control language, it has several other shortcomings that make it ripe for replacement. The motor controllers for the laser are integrated into the board and aren't adjustable, and there's no support for more advanced features like air assist or exhaust fan control. You can't even adjust the laser power with software using the standard board. To some extent this is a safety feature, because you can't accidentally change a software setting and blow up the laser. In reality, it's just another way that the manufacturer can save some money.

TIED UP WITH RIBBONS

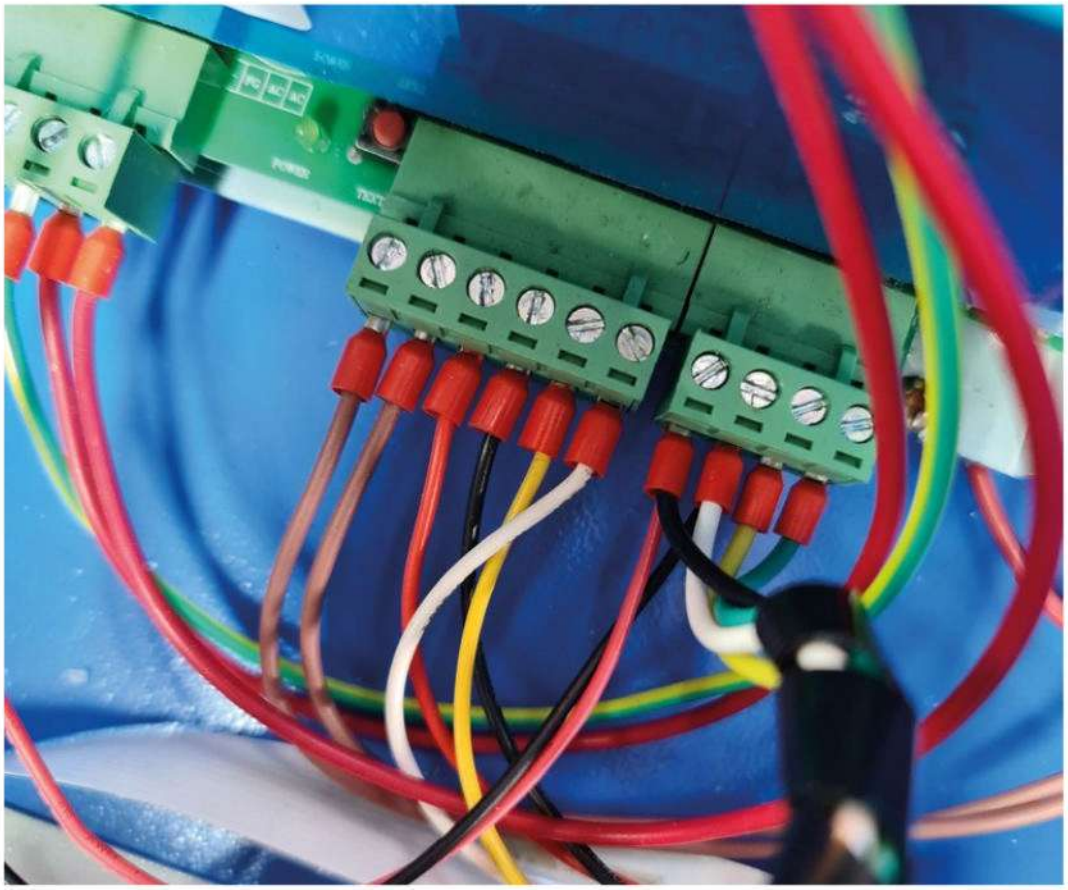
You can use just about any controller to control your laser, and choose between Marlin or GRBL for your firmware, just as you can for a 3D printer. In this project, you'll see that we're using the Awesome.Tech Mini Gerbil v3 (also called the MG3). This board is a fantastic choice to run a K40 for a couple of reasons. There are several variants of the K40, some of which wire the motors to the control board with a conventional cable bundle, while others use a flat ribbon cable for the X-axis motor and endstop. The MG3 supports both types, with a ribbon cable connector built straight onto the board. Most notably, it comes completely preconfigured to run on your K40 with very little extra effort – it's essentially a plug-and-play direct swap for the existing controller, right down to the A4988 controllers being pre-tuned to suit the K40's stepper motors. In theory, if you aren't interested in the extra features of the board and don't care about dead buttons on your control panel, you could replace your existing controller, recalibrate, and be done in about 15 minutes.



Above
The original control board sits vertically at the front right-hand side of the machine, held in place by four bolts. Removing these bolts can be tricky, because the nuts are not fixed on the rear side, but you can usually reach behind with a thin spanner, or even a screwdriver, to hold the nuts in place while you unscrew from the front

AUTO SHUTDOWN

If you've had your K40 for a while and you value your eyesight, you've probably added a safety interlock to the laser chamber door so that the laser switches off when the door is opened. You'll recognise that, although the laser switches off when you open the door, the motors will continue to move, dashing any hope of rescuing a piece unless the program has already been paused. The MG3 board can be used to monitor the laser chamber door, and pause the machine when it is opened. That means the machine will stop working when the door is opened, and you can press a resume button to start the laser cutting again once the door closes. This is a great upgrade, but isn't as straightforward as you might think. Safety interlock systems are designed to be very simple – in the K40, it's essentially a number of switches or relays connected together into a loop. If one of the switches is activated, the loop breaks, and the laser stops working. It's a purely mechanical system with no microprocessor required. From a safety perspective, switching that sort of safety system to software control is a bad idea. Computers can go wrong, and it's possible the system might not behave as expected in an emergency. So how do you resolve this, allowing the convenience of the software pause with the safety of the hardware interlock? One method is to employ multiple switches – one that operates the interlock, and another that connects the LID and GND pins on the MG3 control board. Alternatively, a single DPDT microswitch could perform the same function, with one pole connected to the interlocks, and the other connected to the LID and GND pins of the MG3. Having the microswitch double-throw allows you to decide whether the switch will make or break the connection when the door is opened, to match the system you already have. For more information and discussion about this, have a look at **hsmag.cc/LaserSwitch**.



Left
Throw away your expectations of colour coding for the wires inside a K40. The colour choices appear to be entirely arbitrary. High- and low-voltage wires may even use the same colour and be right next to each other. In this example, both 240V wires are coloured red, and so is the laser active pin, while the 5V is white and also yellow. The laser power wires are black and red, 24V DC is black, and the ground wire is white or black. Chase every wire from end to end – don't trust colour at all

Changing the control board will correct these shortcomings, and as a result, you'll need to make some additional changes to your machine. Most obviously, you'll have a control panel with a lot of buttons on it that don't do anything any more. Laser control will be handled with software, so all of the buttons or dials you have to control laser power will go away. In addition to this, you'll be able to make some safety improvements and upgrades that will make life much more comfortable when using the laser.

Fitting the MG3 control board into the K40 is fairly trivial, and it actually presents a good opportunity to double-check your connections and make sure everything looks healthy inside the machine. It's good practice to take a few photographs of the connections and wires inside the machine before you do anything, just in case anything comes loose while you're rummaging around in there and you need to check where something went.

The control board will have connectors for the X and Y axis motors, input power, and laser control. Disconnect the cables from the control board and throw it onto your scrap pile while uttering a suitably pithy insult. Now turn your attention to the control panel and power switches, and trace the wires that connect to them. You'll probably need to unwrap the plastic cable organiser that is bundling things together and cut a few cable ties to do this. Separate out the high-voltage wires that go to the power and emergency stop switches, and don't mess with them. You're going to leave them connected just as they are, but will disconnect almost everything else from the panel. Most modern K40s have a digital display with push-buttons to control laser power, a rocker switch that operates the safety interlock and prevents the laser from firing, and a switch to turn the light on or off in the laser chamber. None of these switches really make sense for the way that the laser is going to work, and can be removed. A common feature on the K40 control panel is an ammeter to monitor laser power. If you have one of these, you'll need to disconnect it and fit it into your new control panel. The wires that control the push-buttons for the laser power and the test →

> **It's good practice to take a few photographs of the** connections and wires inside the machine

QUICK TIP

Before you disconnect everything and take your laser out of action, cut a replacement control panel.



Above
There are as many designs for K40 control panels as there are K40 laser cutters. A simple Google search will reveal plenty, although it's easy enough to come up with a design that suits your own needs using Inkscape or a similar package

STAYING TRIM

You might find that the first time you run the laser, after changing the board and installing the proper settings into LightBurn, that you'll need to adjust the trim of the motor controller slightly. This is particularly true if you've modified your laser to add air assist or alignment lasers on the gantry. This is because the extra weight of the modified gantry can be enough to overpower the current limit on the A4988 controllers. It's most likely that you'll have an issue with the Y-axis motor, as this is the one that is moving the most weight. Very slightly tweaking the trim pot on the controller by 1/8th of a turn clockwise (or even less) should be enough to correct this. It's tempting, if this happens, to consider replacing the stepper motors with more powerful versions, but this isn't really necessary and will probably just cause more problems. The steppers on the K40 use 0.9-degree stepping, rather than the more common 1.8-degree steppers used by 3D printers. You will need to adjust for this if you replace them. You would also need to add an additional 24 V power supply to avoid overloading the laser supply (which can only output 1A maximum at 24 V). On top of this, if you intend to replace the X-axis motor, you'd probably find that the ribbon cable can't carry enough current to the motor and will need replacing with an alternative cable. To be blunt, it isn't worth the effort in the majority of cases.

button can be removed from the wiring loom. Trace them back to the power supply, and unscrew them from their connectors.

The new control-panel-only template provided here features a couple of buttons to pause and resume the laser, a key switch that replaces the K40's laser enable switch, and the ammeter. There's no practical need for the chamber light to have a switch, so the wires for the light can be connected directly to the 24 V/GND connection of the power supply. If the machine is on, the lights are on.

The pause and resume buttons are normally open, momentary push-button switches that connect between the named pins on the control board and the ground pin. The key switch connects into the interlock loop on the laser power supply, just as the laser enable switch did on the unmodified K40.

With the new control panel fitted in the machine, you're almost done. The power connector for the control board should match the old board and the cable should plug in without any modification. However, it's worth double-checking that you don't have a strange variant of the board and that the pinouts match the new board, just to be sure. Finally, you need to connect the PWM pin on the MG3 controller to the PWM IN or POT IN pin of your laser



QUICK TIP

Using a key as one of your interlocks is a good way to reduce the chance of someone firing the laser without permission, either purposefully or accidentally.

Left LightBurn makes it much easier to perform common laser tasks like automatic shape offsetting, and having multiple cut layers for each project

power supply. There are a few different designs of power supply out there, so you'll need to check the datasheet for your specific model if you're not sure which pin this is. Use the plastic stand-offs that came with your board to mount the MG3 into your case, and double-check all of your wiring before plugging in the power and turning on the machine. Assuming everything looks good, plug in a USB cable and finally connect the laser to your computer.

You should be able to control the laser with any standard G-code controller, but for the best experience, download LightBurn and give it a try. At £50, it's a relatively affordable luxury and it offers some excellent features, even when using the free 30-day trial. On the first run, click on the Devices button and add a new Gerbil-STM device. Follow the setup wizard, setting the dimensions of the work area to 320 × 230, and set the origin of the machine to Rear-Left. Click Finish on the wizard, and your laser cutter should be working with LightBurn and you'll have access to features like automatic shape offsetting, and camera overlays of your laser bed when arranging items.

DOLLAR STORE

The $ settings are G-code's way of letting you fiddle with the internal configuration of your machine to tweak performance and enable or disable certain features. You can assign these values directly in your G-code console by typing something like $96=1 (which would tell the MG4 to turn off air assist when the machine pauses), but if you're using a package like LightBurn, some of these features might be accessible through a graphical menu. Be aware that fiddling with these settings inadvisedly can make your machine do terrible things. Expensive things. Things that go bang and catch fire. Be careful. With that warning out of the way, there are a few settings that you might want to pay attention to if you're using an MG3 on your K40:

$95=1 Inverts the operation of the air assist pin
$96=1 Turns off air assist when the machine pauses
$97=1 enables the laser chamber door safety system
$110=24000 X-axis feed rate (24000=400mm/s)
$111=8000 Y-axis feed rate

A more complete list is available from here: **awesome.tech/what-are-settings**.

Changing motion with linkages

A round of applause for ourselves



Ben Everard

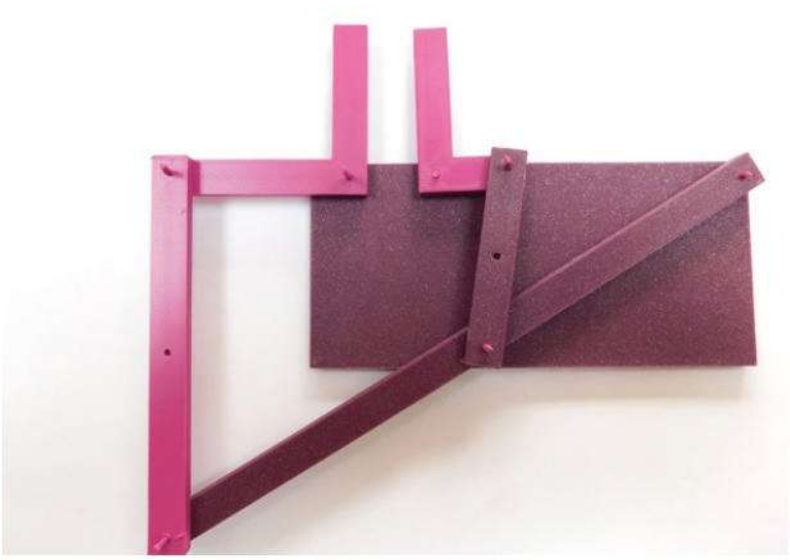
Ben's house is slowly being taken over by 3D printers. He plans to solve this by printing an extension, once he gets enough printers.

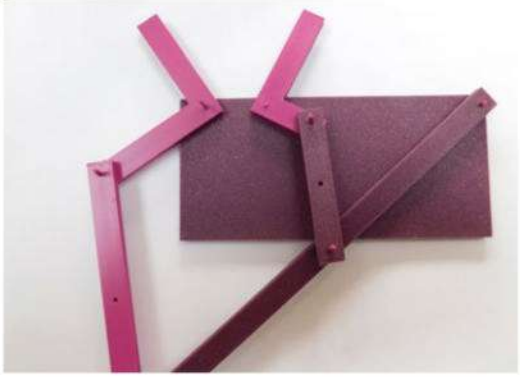
Last month, we looked at linkages – ways of joining together moving parts using solid bars. This month, we're going to design one. We want to create an arrangement of rods and joints that can only move in one way, and that's to make a pair of hands clap. We want both hands to move toward each other.

Alternatively, hands can be on rods that rotate around a point. However, we need one to rotate clockwise and one to rotate anticlockwise.

'Bell (crank) linkage' is a name for an L-shaped rod attached at the angle in the middle of the L. This linkage gives us a way to transfer motion between different directions, because pushing the base of the L up makes the top of the L move sideways. In this way, we can transfer (almost) vertical motion into (almost) horizontal motion. We need to do this for both hands, so we have two Bell linkages. By flipping one around, we can get anticlockwise motion on one linkage and clockwise on the other. This does present a problem because one of the linkages will be further along the arm than the other, and the further along the arm it is, the more motion it will get as the arm moves. We can simply offset this by making the arm on one of the Bell linkages longer than the other.

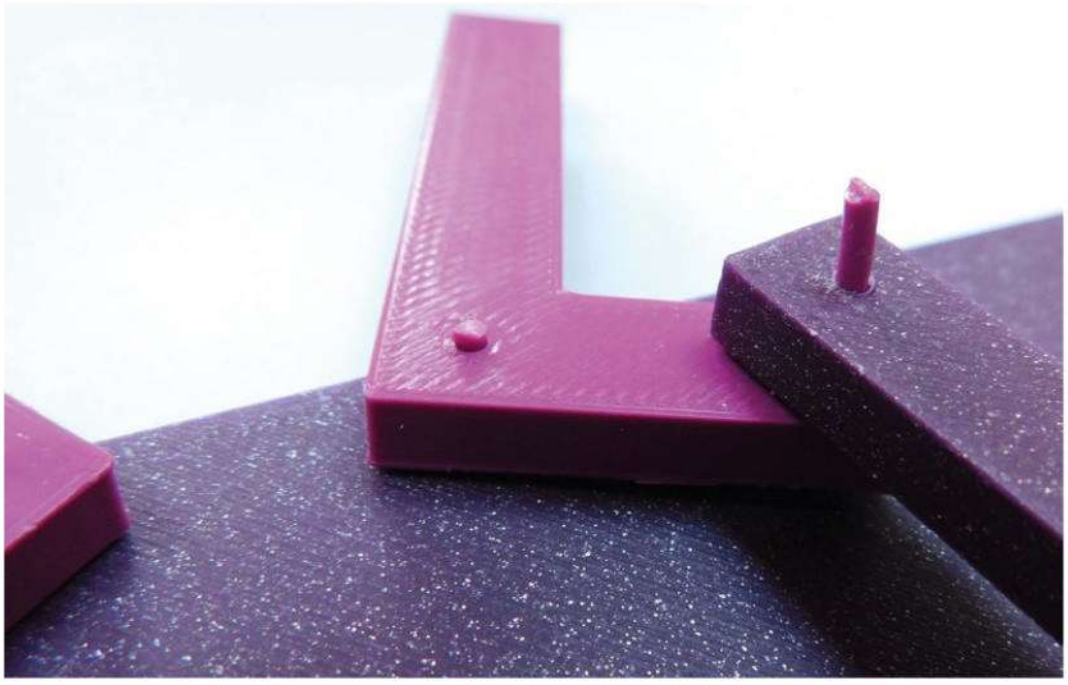
This doesn't quite solve the problem though, as we still need a way to actuate this pair of Bell linkages. We opted for an arm on a pivot. If we attach both linkages to this then, as we rotate the arm, it will move the two Bell linkages. However, one of the Bell linkages will be further out than the other, so will rotate further. The solution to this is to





Above
Open ...

Left
... and shut



Left
Push the filament into the hole, and trim the end for a strong pivot

make one of the Bell linkages have a proportionately longer base to the L, and then the leverage effect counteracts the fact that it moves further.

Once the plan was sketched out, calculating the lengths of everything was just a little trigonometry.

> **Spend a bit of time looking at other linkages and** familiarise yourself with some of the simple ones

DESIGN DECISIONS

As with many linkages, once the design is finalised, everything seems very straightforward. This doesn't mean it was straightforward to design. The best advice we can give for designing linkages is to spend a bit of time looking at other linkages and familiarise yourself with some of the simple ones. Don't forget the importance of CAD – Cardboard Assisted Design. With an old box, some scissors, and some pivots (pretty much anything you have to hand can work, from split pins to screws), you can quickly test out some designs.

Last month, we looked at different techniques for joining our rods together and came to the conclusion that short sections of filament performed well. This month, we've continued to use these. Most of the links are, therefore, bars with 2 mm holes in them at the ends. These are really easy to draw in FreeCAD. We created an object that took data from a spreadsheet and made the bars suitably sized.

The Bell linkage bars are slightly different because these need to have 90-degree bends in them. We created them using two sketches – one with the outline of the bar which we then padded, and one with the holes that we then pocketed.

Finally, you need to add a base, whose sole purpose is to hold everything together.

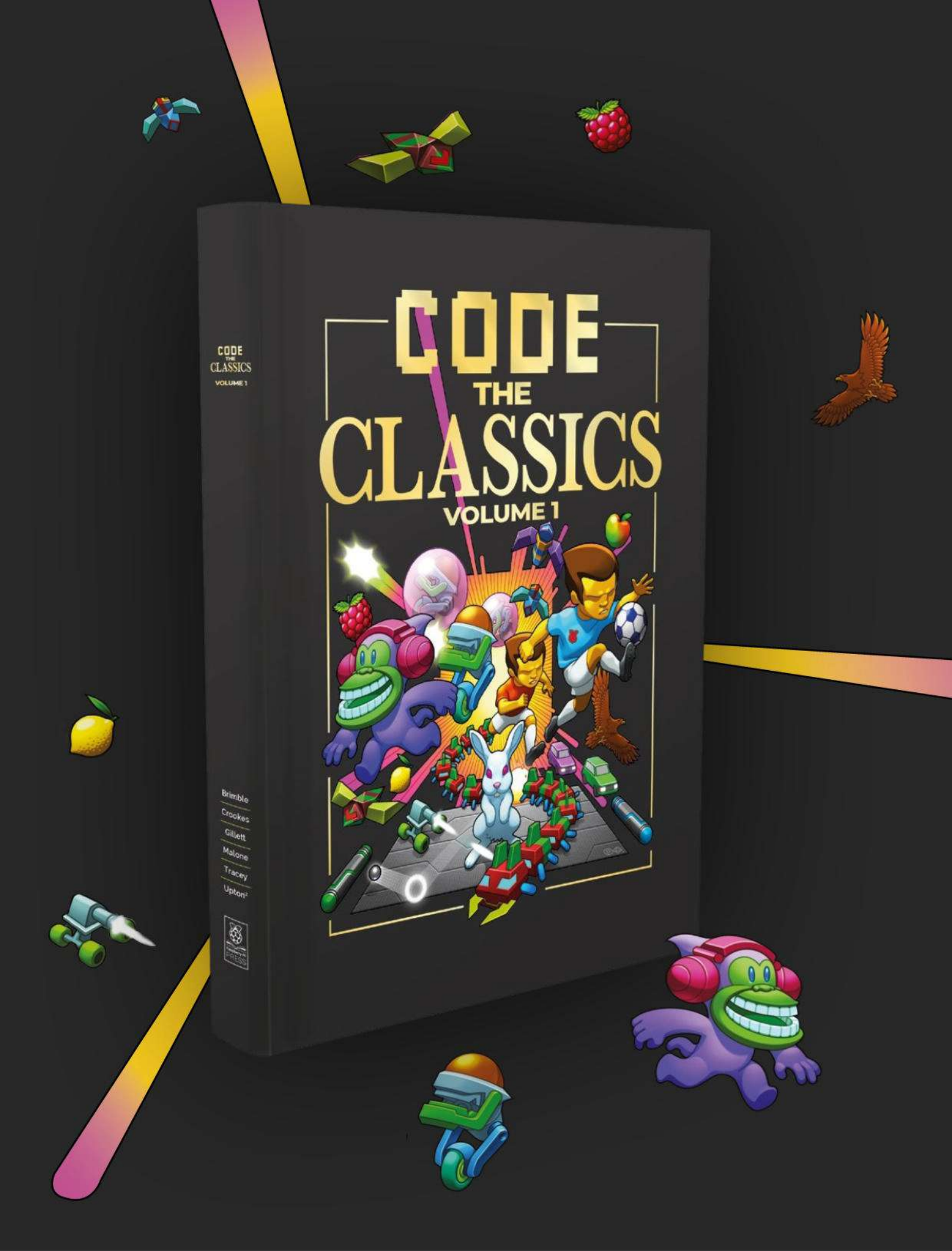
Once we had the linkage set, the only thing left was to add the hands. We traced around our hands and cut them out of cardboard. A bit of sticky tape holds them onto the linkage. Of course, you could print out hands for this if you prefer.

EXTENDING

It would be fairly easy to extend this project by linking the main bar to a wheel that you could turn, either by hand or with a motor. This would make it easy to generate a lot of applause. Equally, you could use the movement to do something else – clang some cymbals together or get a pair of hands to wave goodbye (or hello).



Above
Tape on a pair of hands and clap away


CODE
THE
CLASSICS
VOLUME 1
Brimble
Crookes
Gillett
Malone
Tracey
Upton²





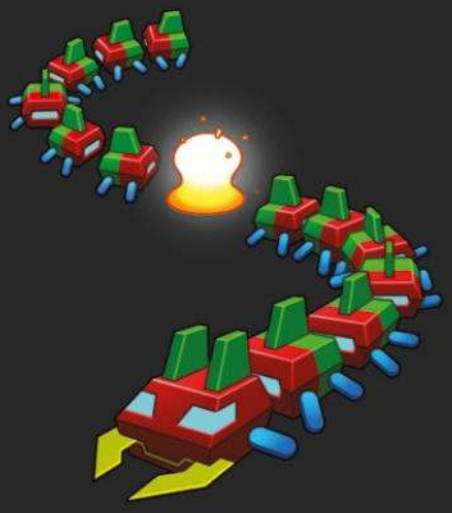
CODE THE CLASSICS
VOLUME 1

This stunning 224-page hardback book not only tells the stories of some of the seminal video games of the 1970s and 1980s, but shows you how to create your own games inspired by them using Python and Pygame Zero, following examples programmed by Raspberry Pi founder Eben Upton.





- ***Get game design tips and tricks from the masters***
- *Explore the code listing and find out how they work*
- ***Download and play game examples by Eben Upton***
- *Learn how to code your own games with Pygame Zero*



Available now **hsmag.cc/store**

Pico Agile Octopus Tariff Display

Get live updates of your electricity prices



Rob Miles

Rob Miles has been playing with hardware and software since almost before there was hardware and software. You can find out more about his so-called life at **robmiles.com**.

Find out how to make a Raspberry Pi Pico W-powered electricity tariff display to help plan your energy use. Along the way, you will also discover how to configure settings in an embedded device using your browser.

AGILE ENERGY TARIFFS

If you are looking for a way to reduce your electricity bills, you might want to look at the world of agile pricing. The author is presently using an agile tariff and it seems to be working so far, making noticeable savings. The author doesn't have any solar panels or batteries at his house, but he does drive an electric car. He doesn't have any special control systems for any of his household devices; the savings have been achieved just by doing things at different times. He is using the Agile Octopus tariff (**hsmag.cc/AgileTariff**). There may be similar offerings from other suppliers.



Figure 1 Sometimes the price goes negative, and you get paid for using electricity. Unfortunately, this doesn't happen very often (although it did on this day at 5:00 am)

SLICED DREAD

The Agile Octopus tariff divides the day into 48 half-hour 'slices', each with its own price. The prices for the slices (sorry) are released one day in advance. They are calculated by Octopus from the projected cost of energy for that time. If the weather is going to be windy and sunny, there will be a lot of cheap, green energy available, and prices will go down. On cloudy, still days, the price may go up. During the day the price varies, usually with a couple of peaks (a smaller one in the morning and a larger one in the evening). The peaks are very predictable (the largest one starting around 4:00 pm) and only during the middle of the peaks does the price exceed the price the author was paying on his previous tariff.

Figure 1 shows the display from the Agile Octopus website, giving prices over the day. The author was surprised to discover that night-time prices (which are usually associated with off-peak energy) are not always lower than daytime ones (although they are in **Figure 1** – it must have been a windy night). It will be interesting to see how prices change during the winter months, with more wind and less sunshine.

TARIFF DISPLAY

You can get the tariff information on your phone or computer, but the author really wanted a gadget that could be kept in the kitchen (where a lot of the power is consumed) and help decide when to do things. The tariffs are exposed by Octopus as JavaScript Object Notation (JSON) formatted data which is read by a Pico W connected to the home Wi-Fi. The author has been through a few design iterations with his wife (who is playing the role of the customer for this development).

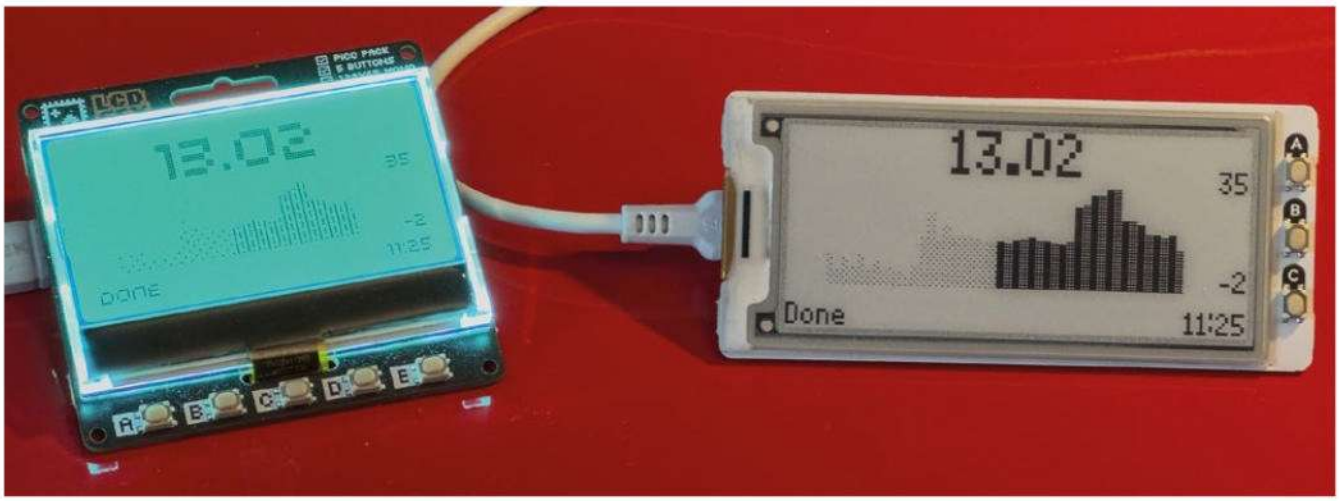


Figure 2 shows the two versions of the display. The left-hand display uses a backlit LCD panel, and the right-hand one uses e-ink. Both devices are showing the same data. The number at the top gives the current price per kilowatt hour. An electric kettle uses around 3 kilowatts, so if you spent an hour making tea at the price shown this would cost you around 39 pence (3*13). The dark bars show upcoming prices. Each bar represents the price for that time slice – the solid bars are hourly prices, and the dotted bars are half-hourly. The numbers on the right-hand side of the display give the maximum and minimum prices for the day. The maximum price for the day (which occurs at around 6:30 pm) will be 35 pence. The minimum price will be -2 pence. This means that for the magical time slice (from 5:00 to 5:30 am) you would get paid for consuming power.

> For the magical time slice – from 5:00 to 5:30 am – **you would get paid for consuming power**

The faintly dotted bars show the previous prices for the day. The time is shown in the bottom right-hand corner. The left-hand corner shows status messages.

This display has been in use for a while and is useful for quickly deciding when to turn on the washing-machine or tumble dryer, and when to bake a cake. It is powered by a Raspberry Pi Pico W running a MicroPython program which gets and displays the tariff values. You can find the code on the GitHub site for this project at **hsmag.cc/AgileInstall**.

GET YOUR TARIFFS HERE

Octopus provides web addresses for all its tariffs, which are exposed as JSON-encoded files. You can get the tariff code from your Octopus account details. Visit your Personal Details page and select the API (application programmer interface) page. This shows your unique key that you can use to validate requests to read your energy usage data from Octopus. If you scroll down the page, you can find the item marked 'Unit rates' as shown in **Figure 3**. The address we will use is the part that hasn't been crossed out.

HARD TIMES

One of the biggest problems the author had with building the display turned out to be dealing with time, more specifically, British Summer Time. This moves the clocks in the UK forward in summer. Unfortunately, the items in the data supplied by Octopus are not adjusted for this change, so during summer, the display must ask for readings starting one hour before the required time. This means that the display must decide if it is in summertime before asking for tariff values.

```
period_from=2023-09-22T23:00Z&period_
to=2023-09-23T23:00Z
```

The code above shows the time period that was fetched by the display for the 23 September. Because this is in summertime (one hour ahead of the time standard used by Octopus), the display fetches values from one hour behind the current time, which means that the first hour is fetched from 'yesterday'. Fortunately, in a previous life, the author wrote a lot of code for managing date stamp production and was equal to the task. The program contains a TimeManager class which does this.

Figure 2
We are not sure if the price values are being shown too precisely. Perhaps we don't need to see fractions of a penny

YOU'LL NEED

- **A suitable box** (some 3D printable designs are also available)
- **A Raspberry Pi Pico W** (remember to get the W version as this has Wi-Fi capability)
- **A display.** The software works with either liquid crystal (Pico GFX) or e-ink (Pico Inky) devices from Pimoroni
- **A smart electricity meter.** The agile tariffs can only be used if your meter can send half-hourly readings to your electricity provider. The author's lowly SMETS1 meter proved equal to this task, but this does vary from meter to meter and supplier to supplier



Figure 3
If you open the web address in your browser, you will see a page of JSON which contains the current tariff values

BUILDING AN ADDRESS

The web address can be followed with query values to specify the start and the end of the period for which prices are required.

Figure 4 shows a complete address that would obtain the price data for 23 September 2023. The MicroPython program controlling the display uses the current date and time to build a string like this.

SAY HI TO THE WI-FI

Before any data can be fetched into the Pico W, the program must first connect to the local Wi-Fi network.

```
import network

self.wlan = network.WLAN(network.STA_IF)
self.wlan.active(True)
if not self.wlan.isconnected():
    self.wlan.connect(ssid, pwd)
    while not self.wlan.isconnected():
        pass
```

The MicroPython code above sets up a connection to the local Wi-Fi using the values in **ssid** and **pwd** as the access point name and password. Once the display has connected to the local network, it now needs to obtain the current date and time so that it can request the tariff data for the day.

```
import ntptime

ntptime.settime()
```

The code above sets the time in the Pico W from the network. In the finished application, the code that connects to the network and sets the time is part of a **Connection** class which manages the network connection.

FETCH THE DATA

It turns out that it is very easy to fetch the tariff data once the Pico W has been connected to the local Wi-Fi. The **urequests** library can do this with a single function call:

```
import urequests

res = urequests.get(url+"?"+period)
```

The code above creates a tariff request containing the URL and the period query values and then calls the **get** function in **urequests** to get the tariff values into a string called **res**. This data is supplied as a JSON-encoded string:

```
{
    "count": 46,
    "next": null,
    "previous": null,
    "results": [
        {
            "value_exc_vat": 17,
            "value_inc_vat": 17.85,
            "valid_from": "2023-09-23T21:30:00Z",
            "valid_to": "2023-09-23T22:00:00Z",
            "payment_method": null
        },
        {
            "value_exc_vat": 17.2,
            "value_inc_vat": 18.06,
            "valid_from": "2023-09-23T21:00:00Z",
            "valid_to": "2023-09-23T21:30:00Z",
            "payment_method": null
        },
        ... another 44 values are here
    ]
}
```

Above you can see the JSON returned from the web request. Only the first two of the 46 price values are shown. The **next** and **previous** items are used if the response forms part of a larger response to a request

LCD VS E-INK

The author wasn't sure whether to use a liquid crystal (LCD) panel or electronic ink (e-ink) for the tariff display device. Fortunately, the displays weren't particularly expensive, and so he decided to get one of each and see which looked the best. The picture in **Figure 2** doesn't really do the LCD panel on the left justice. For an LCD device, the screen is actually very good. It also has a multicoloured backlight, so it would be possible to make the display light up red when power was expensive. The e-ink display looks very nice, but it does take a couple of seconds or so to update when it is redrawn. During the update, it flashes the whole screen black and white, which can be distracting.



Figure 4
The query (?) character in the URL string marks the start of the query values which specify the time range

for a lot of data. This response contains only 46 items, so next and previous are not needed. The **results** item is a list of price values for different time periods.

```
results = res.json()["results"]
```

The MicroPython statement above decodes the JSON and creates a variable called **results** which holds all the result values in a list. The program can then iterate through these values, find the highest and lowest prices and use the **price** and **valid_from** elements to build a list of all the prices.

```
for result in results:
    price = result["value_inc_vat"]
    valid_from = result["valid_from"]
    .. build the histograms
```

The program draws the bar chart display differently for the e-ink and LCD versions. The e-ink display is made from shaded boxes, whereas the LCD version is made from lines. This is done to get the best results on the two different displays.

CONFIGURING WITH THE BROWSER

When you create an embedded device, you sometimes need to modify the program code to set up things like Wi-Fi passwords and other device specific data. The author wanted to avoid this with the tariff display. He wanted it to be possible to configure the display without changing the code in it. **Figure 5** shows how this is done. The display owner connects their device to their computer using a USB-serial connection (the same one that was used to load the software onto the Pico). They then open the configuration page in the browser on their computer. The webpage contains JavaScript code which updates the settings in the device.

Figure 6 shows the webpage which is used to configure the display. JavaScript code in the page communicates with the display and generates this configuration page which is used to enter setting values. These are then sent back to the display which stores them in local filestore on the device. You could use this to manage any MicroPython-powered device that contains user-defined setting values. You can download the configuration site from GitHub, or you can use it online at **hsmag.cc/AgileGitHub**.

CREATIVE LAZINESS

A good description of a programmer is 'creatively lazy'. Programmers build things which make their lives easier. The web-based configuration of the display is a good example of how creative laziness can save a lot of work. The author could have created a custom webpage for the display containing the elements you see in **Figure 6**. However, he decided to create a system where the page creates itself from the settings information. This approach uses the ability of JavaScript code to generate new webpage elements from software.

The settings are stored in the display device as a string of JSON. The string describes an array containing setting values. This string is kept in a file in the display device and sent to the webpage when it is being configured.

```
{
    "wifiSSID": {
        "name": "WiFi SSID",
        "desc": "Name of the SSID point for the
network",
        "type": "text",
        "value": "",
        "order": 1
```

Figure 5
The settings for the display are stored in a text file on the display



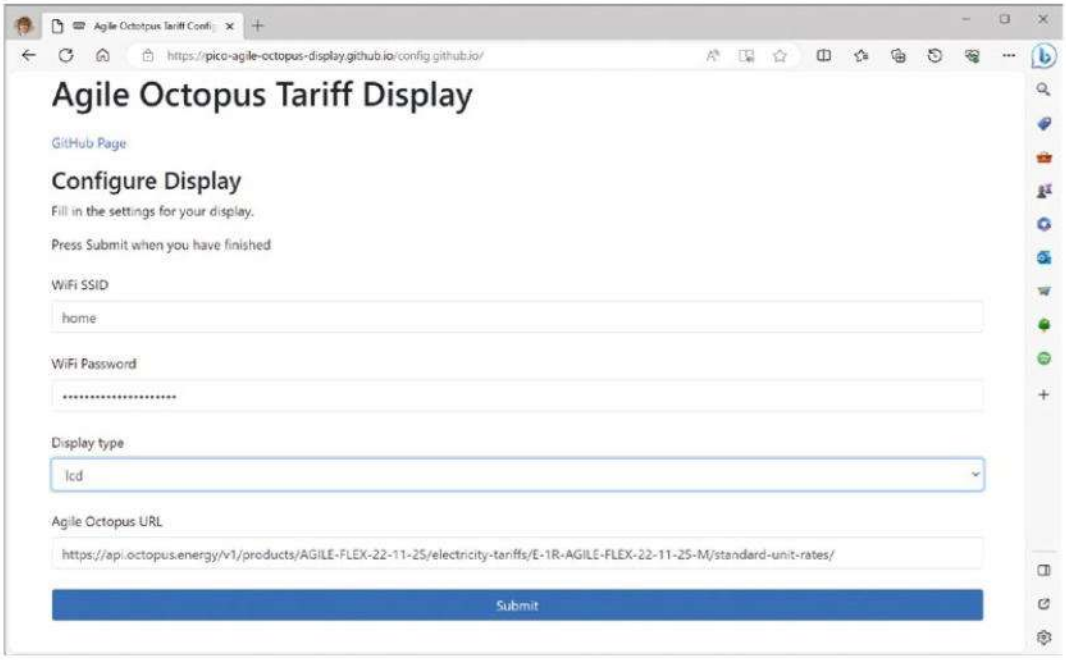


Figure 6
The setting for the display type is selected from the options displayed on the page. You can select a 'print' display which prints values on the MicroPython console for testing

```
    },

    "wifiPWD": {
        "name": "WiFi Password",
        "desc": "WiFi password for the network",
        "type": "password",
        "value": "",
        "order": 2
    },
    "display": {
        "name": "Display type",
        "desc": "Type of display being used in
the device",
        "type": "string",
        "value": "lcd",
        "values": ["print","e-ink","lcd"],
        "order": 3
    },

    "AgileURL": {
        "name": "Agile Octopus URL",
        "desc": "URL for the Agile Octopus price
data",
        "type": "text",
        "value": "https://api.octopus.energy/v1/
products/AGILE-FLEX-22-11-25/electricity-tariffs/
E-1R-AGILE-FLEX-22-11-25-M/standard-unit-rates/",
        "order": 4
    }
}
```

Above are the setting values for the display. Each setting has the properties' name, description, type, value, and order. The order property is used to determine the order of the items in the page created by the browser. The type property determines the type of HTML element used to edit the setting on the webpage. The display property has a values element which gives the possible values it can have.

BUILDING WEBPAGES FROM JAVASCRIPT

When you visit a webpage in your browser, there is a good chance that the page content that you see is not fetched from a web server. Instead, the webpage you receive will contain a JavaScript program which runs in the browser. This code will fetch a small amount of data from the server and use this to create the page that you see. This is how our tariff display configuration works. The configuration page contains a single HTML element:

```
<p id="stageDescription"></p>
```

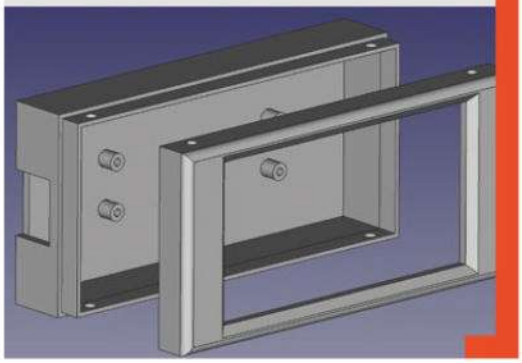
The element above describes an empty paragraph. The element has been given the identifier

SECURITY IS A THING

When you create an embedded device, you must be mindful of security issues. The tariff display stores Wi-Fi settings which can be extracted from it. Anyone can connect a USB cable to a MicroPython device and read back the contents of files on the device. The author considered encrypting the settings in the tariff display, but this would not provide a completely secure solution because the device must also decrypt them before they are used, so the data will always be obtainable by a determined hacker. In the end, it was decided that the best solution was to connect the tariff display to a guest Wi-Fi account on the home network. If the display is lost or stolen, the guest password will be easier to change without affecting other devices.

MAKING A CASE

After a design review meeting in the kitchen, it was decided that the e-ink display was to be used. The case design is shown below. The design box was produced using a Python program running inside FreeCAD to make the initial designs. The lid was then tweaked with the addition of chamfers and bevels. The design files are available on the GitHub site for the project.



stageDescription. When the configuration page displays the setting values, a JavaScript program runs, which first gets a reference to this element:

```
let stageElement = document.
getElementById("stageDescription");
```

The statement above asks the document object model (DOM) managed by the browser for a reference to the **stageDescription** paragraph. HTML elements can act as containers for other elements. The program is going to add extra elements to this paragraph, at which point they will appear on the browser page. The JavaScript variable **stageElement** refers to the empty paragraph on the page which is going to be filled with the input elements.

The program can now add all the elements that are needed to edit the settings of the display to this element. In the code below, the variable **settingsObject** contains a JavaScript object which was created from the JSON settings description received from the display:

```
settingsObject.forEach((setting) => {
    let id = setting[0];
    let value = setting[1];

    inputElement = document.
createElement("input");
    inputElement.setAttribute("type", value.
type);
    inputElement.value = value.value;
    stageElement.appendChild(inputElement);
});
```

The code above loops through the items in **settingsObject** and uses the document.**createElement** method to create an HTML input element for each setting. The type and the value of this element are set to the values for the setting. Then the new element is appended to **stageElement** using the **appendChild** method. At this point it will appear on the page.

The actual code is slightly more complex than the above, in that a label element is also created, but you can see the underlying principle at work here. The result is that the contents of the page is determined by the JSON setting description received from the display device. This is 'creative laziness' in action (or should that be inaction?) in that if we add new settings to the JSON in the display, we don't have to do anything else. The new settings will automatically appear on the configuration page.

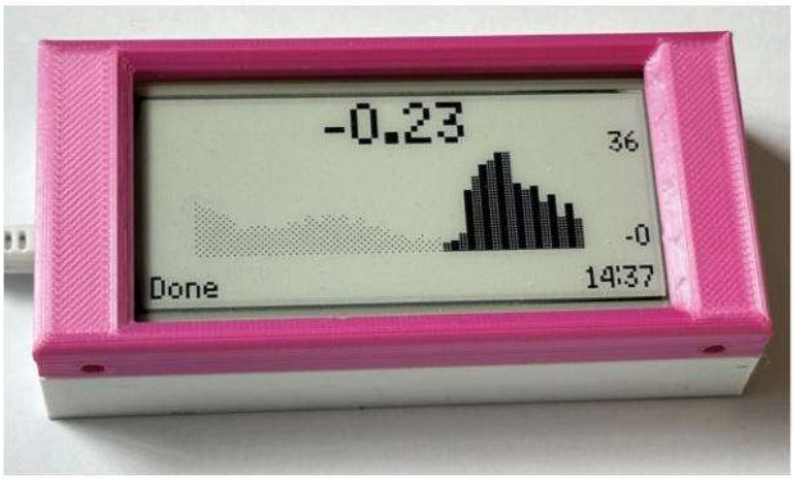
When the Submit button is pressed, another JavaScript function runs with extracts the values from the elements, builds a JSON object, and then sends it back to the display device.

FURTHER DEVELOPMENT

The displays have buttons which could be used to select different display options. The Octopus API allows customers to read back their usage, so it would be possible for the display to show how much electricity was used in each of the time slices. This would be a great extension to the project. But as it stands, it has already proved effective.

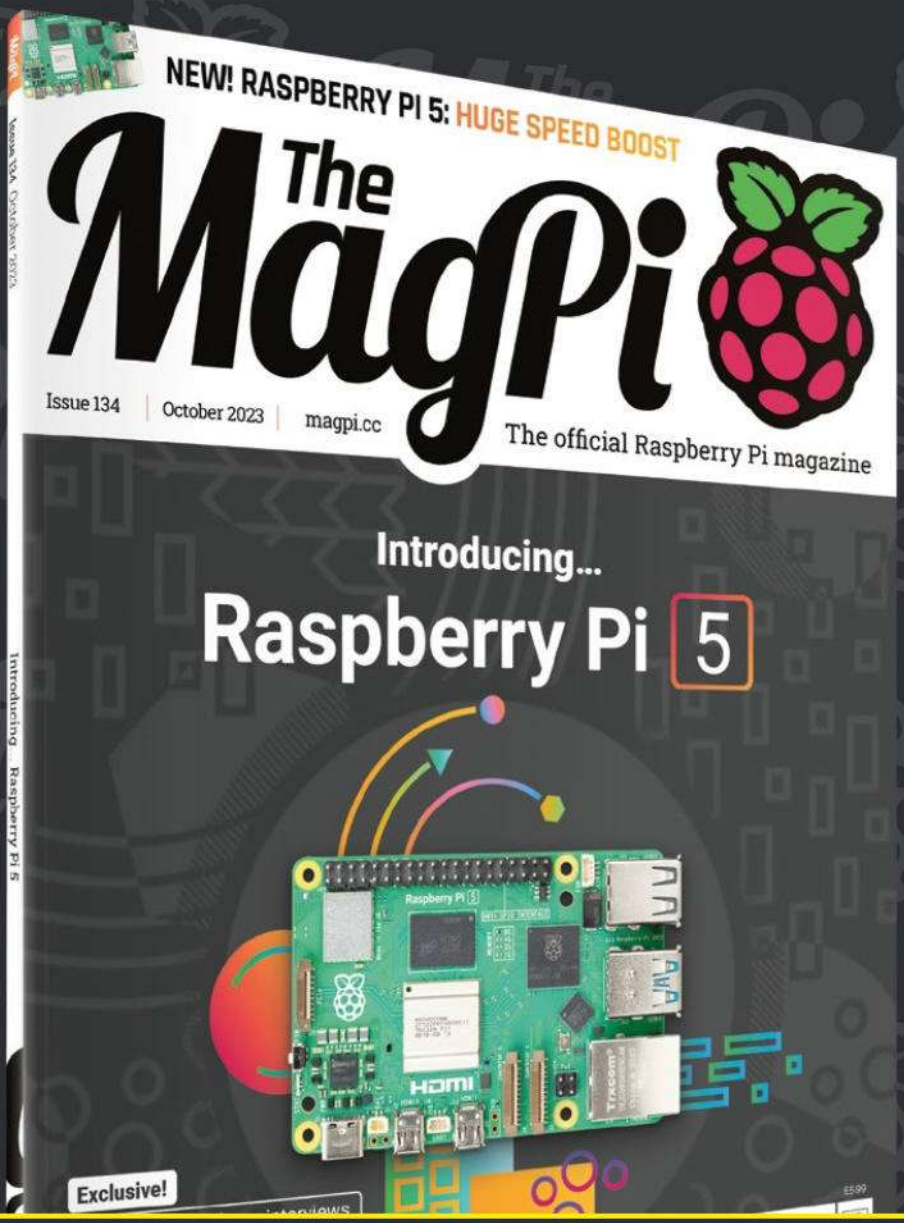
QUICK TIP

If you are happy to modify the MicroPython program in your display, you just have to edit the DEFAULT_SETTINGS string in the Config class.



Below
The lid of the display was printed the same way up as you see it here. This gave the best possible finish to the edges and the display surround

DON'T MISS THE **BRAND NEW** ISSUE!



SUBSCRIBE FOR JUST £10!

- **THREE!** issues of The MagPi
- **FREE!** Raspberry Pi Pico W
- **FREE!** delivery to your door



+ FREE RASPBERRY PI PICO W*

Three issues and free Pico W for £10 is a UK-only offer. Free Pico W is included with a 12-month subscription in USA, Europe and Rest of World. Not included with renewals. Offer subject to change or withdrawal at any time.

magpi.cc/subscribe

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

PG 88

INPUT SHAPER

Prusa printers get a speed boost

PG 90

LIBREPCB

A new way to design your circuits

PG 94

KALUMA

Program your Pico with JavaScript



PG 82

BEST OF BREED

Bar codes are the best codes



ONLY THE BEST

Bar-code readers and more

Sensors for reading the messages found in bar-codes and QR codes

By **Marc de Vinck** @devinck

Being able to read bar-codes isn't just reserved for big box stores for quickly looking up a price, or giant warehouses tracking the flow of their inventory. There are plenty of other use cases for using bar-codes; even small electronic kit makers can use them to track limited inventory and the steps in the manufacturing process. I've also seen some really creative uses for bar-codes and QR codes that include interactive art, musical instruments, and even as a discrete set of instructions for robotics. There are plenty of small manufacturers out there that need a custom solution where the typical handheld scanners available just don't work.

This is certainly a very niche topic for this month's Best of Breed, but I found the whole ecosystem of how they work, and the multitude of types of QR and bar-codes out there, kind of fascinating. And since there are dozens, if not hundreds, of different inexpensive handheld scanners available, it only seems natural that you'd be able to buy the discrete components to build your own. My adventure started when I found a very interesting and affordable QR code scanner from Adafruit. So, let's jump right in and take a look.

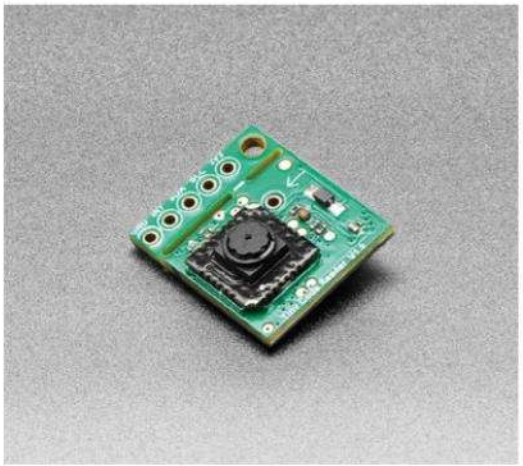


Tiny Code Reader from Useful Sensors vs Zero Barcode HAT

ADAFRUIT $7 | adafruit.com **TINDIE** $69.99 | tindie.com

This is the product that inspired the entire roundup. The Tiny Code Reader from Useful Sensors is available at Adafruit. It has an RP2040 on the back and a camera module on the front. It allows you to quickly scan a QR code and feed the data into a microcontroller or single-board computer. And it's really inexpensive! Just $7!

The on-board microcontroller runs TensorFlow Lite and does some complicated image recognition to decode the QR code. Simply connect it to your microcontroller via the STEMMA QT / Qwiic connector or wire it up the old-fashioned way. It's amazing to think about this $7 technology compared to what was available ten years ago. We are in the future!





Next up is the Zero Barcode HAT from SB Components in the UK. This HAT is the perfect addition for your Raspberry Pi Zero. It features a DE2120 scanner module, buzzer, 1.14" LCD screen, and a micro USB connector to get you up and running fast. It can scan any of 20 different bar-codes, both 1D and 2D. And since it's a camera, it can also decode QR codes. Once scanned, you can display the data on the crisp LCD screen. It's a complete system, minus the Raspberry Pi!

The HAT also has two LEDs: one for lighting up the graphic, and the other to project the red line, like laser-based scanners. The data the sensor reads can be transmitted via the on-board micro USB port or the UART pins. If you have a Raspberry Pi Zero and want to include some bar-code scanning abilities, this is a great HAT for your project.

VERDICT

Tiny Code Reader from Useful Sensors

Amazing tech at an amazing price.

10/10

Zero Barcode HAT

Capable, but at a price.

8/10

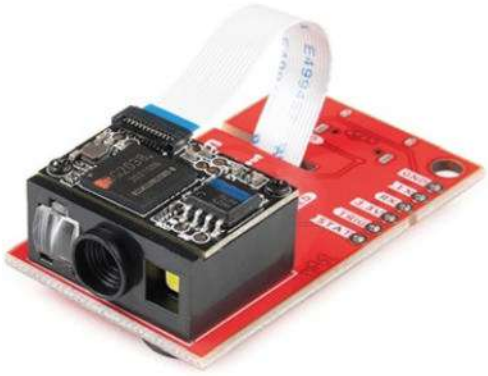
BEST OF BREED

SparkFun 2D Barcode Scanner Breakout

SPARKFUN $53.50 | sparkfun.com

The SparkFun 2D Barcode Scanner uses the same DE2120 sensor as the Zero Barcode HAT, so it has many of the same functions. However, this breakout board isn't in the Raspberry Pi Zero form factor, and it doesn't have a screen, so you save a little bit of money if you don't need those features. But it's still very Raspberry Pi-friendly!

This breakout board aims to make it easy to access all the features of the DE2120 without dealing with its small pitch and sometimes finicky flat flex cables. You'll have basic feedback thanks to the on-board LED and buzzer. Check out the comprehensive SparkFun 2D Barcode Scanner Breakout Hookup Guide for a lot more information.



VERDICT

SparkFun 2D Barcode Scanner Breakout

A breakout board for a reliable sensor.

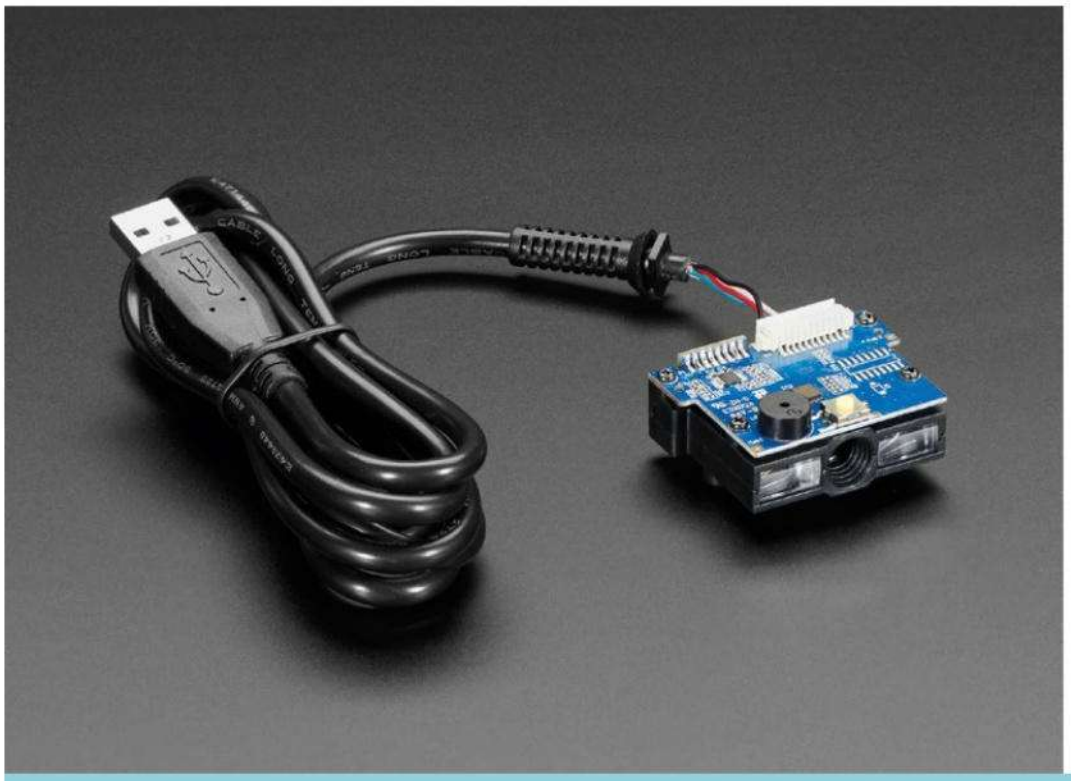
8/10

RFID WIZ KIT BY SMOOTH TECHNOLOGY

ADAFRUIT $49.94 | adafruit.com

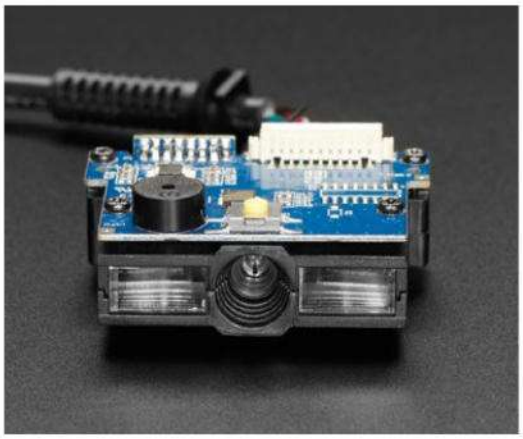
Yes, I know, this isn't a bar-code or QR code reader. But when researching the topic, I found this handy little RFID reader kit available at Adafruit. What I like about this one, compared to a whole host of other versions, is the built-in relay and the fact that it's a complete kit with no code required. The kit includes the RFID Wiz board, 13.56MHz Tag Reader board, 12V 2A power supply, 6-pin JST-XH cable, 2-pin JST-XH cable, RFID key fob, RFID card, 5 × RFID stickers, and a QR code that directs to the Adafruit learning guide. Simply train the board with the included RFID cards and start making!





Barcode Reader/Scanner Module - CCD Camera - USB Interface

ADAFRUIT $69.95 | adafruit.com



The Barcode Reader/Scanner Module from Adafruit can decode nearly any kind of 1D bar-code in a very small form factor. The sensor takes 100 photos per second and is very similar to the laser-based scanners, but with no moving parts, so it's more robust.

All you need to do is plug it into any available USB port and it will show up as an HID keyboard. When you scan a bar-code, the raw data is decoded and parity-checked, then output to your computer as a series of digits, just like entering them on a keyboard. And if you don't want USB output, check out the PS/2 interface version which is compatible with almost any microcontroller.

VERDICT

Barcode Reader/Scanner Module - CCD Camera - USB Interface

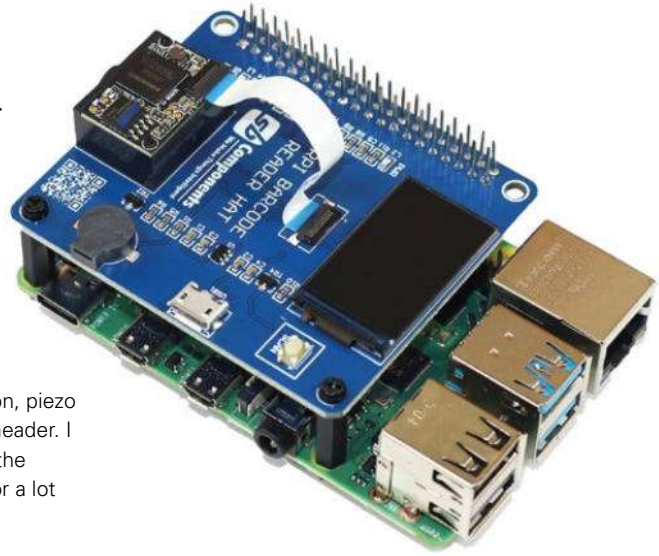
Compact and ready to go.

8/10

BEST OF BREED

Barcode HAT for Raspberry Pi 1D/2D Codes Reader

TINDIE $76.94 | tindie.com



Once again, we find the ever-popular DE2120 scanner module, but this time in a form factor that's perfect for your Raspberry Pi. The Barcode HAT for Raspberry Pi by SB Components is a perfectly sized and capable bar-code scanner. In addition to the DE2120, the board features a 1.14" LCD screen for displaying scanned results, a micro USB port, LED, button, piezo buzzer, and handy pass-through 40-pin GPIO header. I like the fact that you can still access many of the GPIO pins after attaching this HAT, allowing for a lot of expansion possibilities.

VERDICT

Barcode HAT for Raspberry Pi 1D/2D Codes Reader

Form-fitting for your Raspberry Pi.

9/10

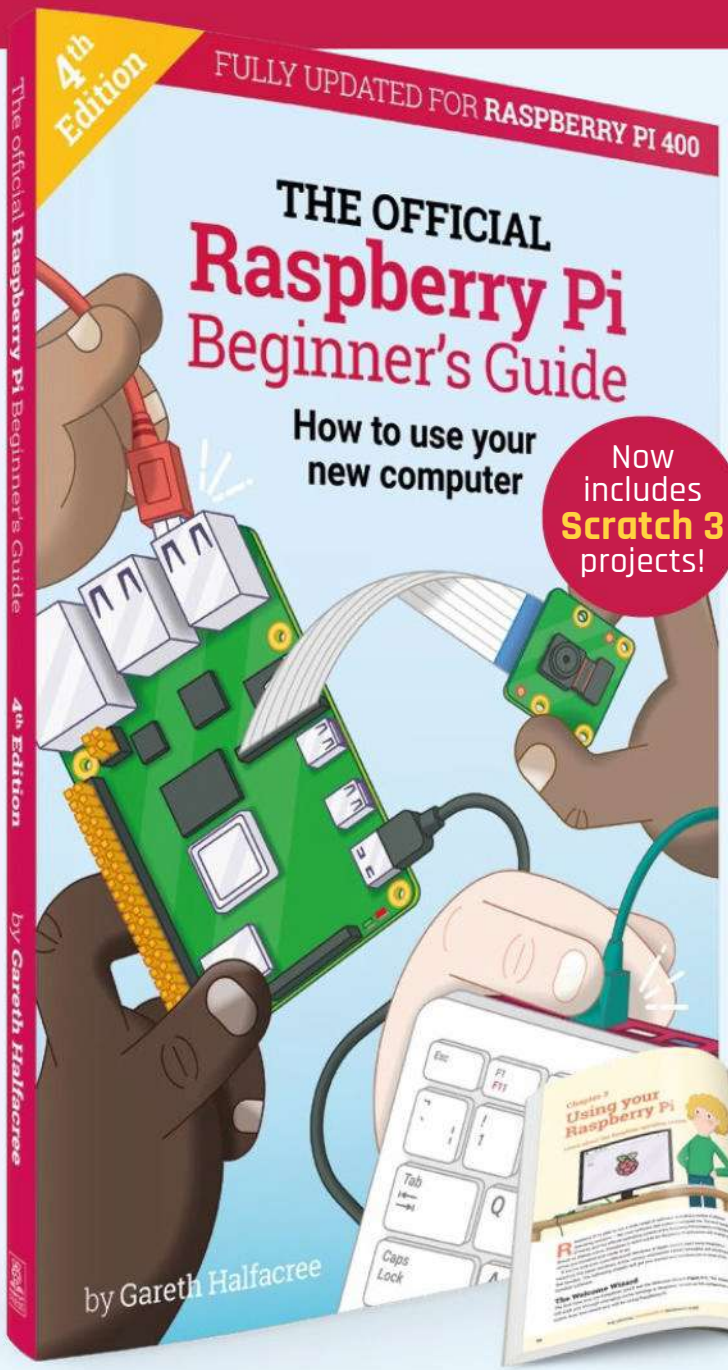
HANDHELD 2D & 1D USB BARCODE SCANNER

PIMORONI $47.93 | pimoroni.com

OK, so you aren't into building a bar-code scanner, we won't judge. But when you start looking at what is available out there, it can be overwhelming. That's where this USB scanner comes into play. Pimoroni offers a simple, easy-to-program USB scanner. Once you figure out how to get access to the built-in USB cord (yes, there is a trick), you can simply scan the bar-code on its site to change the functionality from factory default to a much handier USB HID controller or a variety of other possibilities. Head over to its website to learn more.



THE OFFICIAL Raspberry Pi Beginner's Guide



The only guide you need to get started with Raspberry Pi

Inside:

- Learn how to set up your Raspberry Pi, install an operating system, and start using it
- Follow step-by-step guides to code your own animations and games, using both the Scratch 3 and Python languages
- Create amazing projects by connecting electronic components to Raspberry Pi's GPIO pins

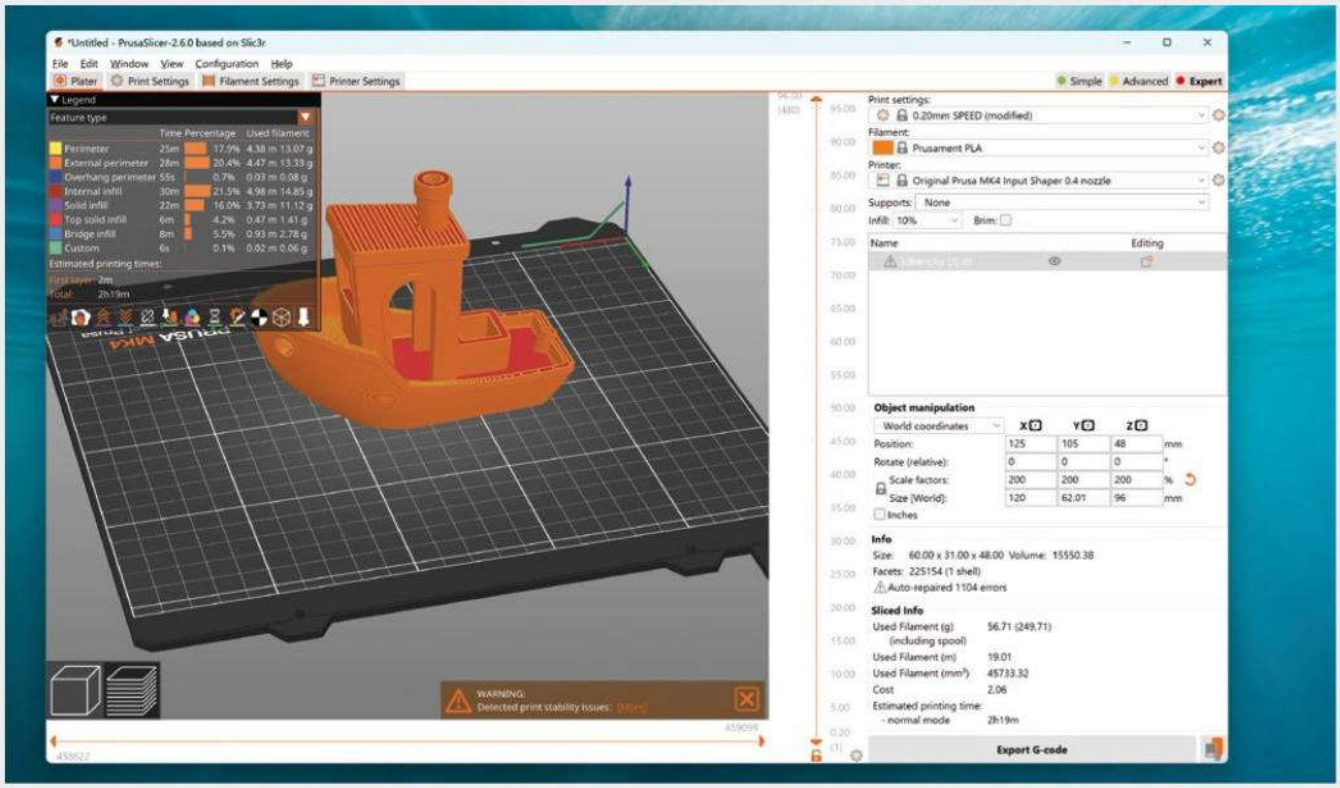
Plus much, much more!

Just £10



Buy online: magpi.cc/BGbook

REVIEW



Prusa MK4 input shaping

Get more speed out of your printer

FREE UPGRADE | **prusa3d.com**

By Ben Everard

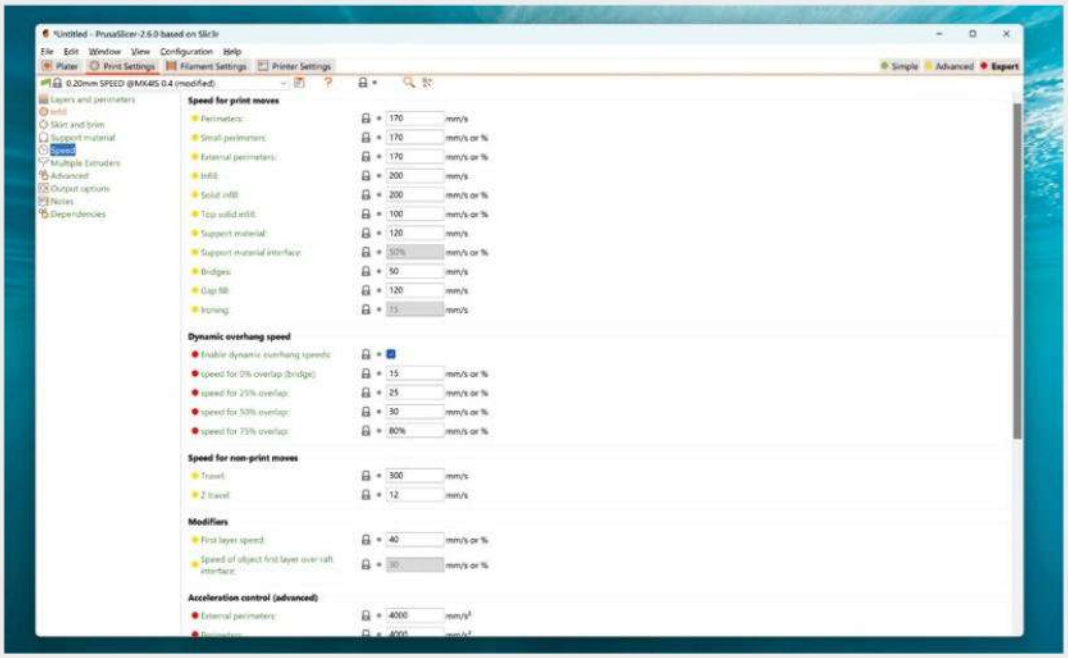
Back in issue 67, we reviewed the Prusa MK4. At launch, we found it to be a good printer, but it wasn't quite as fast as some of the competition. Lingering behind that was the promise of improved firmware to speed it up.

The limiting factor in how fast you can 3D-print is usually not the actual speed the motors can move, or even how fast the processor can go, but how much the frame shakes. Because 3D printing typically involves lots of short moves, increasing the speed only really has an effect if you can increase the acceleration and deceleration as well. To a certain extent, you can mitigate this by making the frame stronger. However, increasingly, the most effective way of mitigating this is in software.

If the processor on the printer can predict how the frame will move and wobble at a particular amount of force, it can compensate for this in how it moves the motors. This is what input shaping refers to. The latest version of firmware for the MK4 printers includes this.

They are by no means the first printers to include input shaping, but they have implemented their own version of it.

Above
It's Benchy, but bigger



Left
At these speeds, everything starts to shake, so make sure your printer's on a solid table

IN PRACTICE

We've been using the input shaper in both its pre-release alpha version and the release version for a while now, and it's working well. We're basically getting the same prints in less time and there's not much to complain about there. Exactly how much faster it goes depends a lot on the specific model, and the quality you're after. The classic Benchy, for example, takes 38 minutes when printed at 0.2 mm layer height and sliced normally. If you really want to push things, you can print a pretty decent-looking Benchy in twelve minutes if you're willing to sacrifice good printing practice (take a look at **hsmag.cc/BenchyInputShaping**).

However, while Benchy is a great test for a lot of things, it's not actually that indicative of the sort of speed changes you're likely to see because it's a small, intricate model. To give you a better idea, we scaled Benchy up to 200% of its size. At this scale, it's a bit more like most of the prints we do (though your usage may be different). A Prusa MK4, with input shaping, prints in 2h 19m. Without input shaping, it's 3h 54m. On a Prusa MK3, it's 4h 59m. This is broadly indicative of the times we've been seeing with real-world models.

This sort of speed makes it comparable with the fastest alternative consumer 3D printers. Which printer will be fastest for you will depend on the particular model you're printing and the slicer settings you use. However, there's not going to be a significant difference one way or the other. More exotic machines (such as homemade Vorons) can end up faster, but that's a whole different class of printer.

> **This sort of speed makes it comparable with the** fastest alternative consumer 3D printers

Perhaps the first thing you'll notice when you start using the new firmware is that the names of the slicing profiles have changed. There's no longer 'speed' and 'draft', now it's 'speed' and 'structural'. They should both have a similar look, but the faster prints can be a little weaker, so if you need your prints to stand up to the tests of time, you'll want structural.

Currently, this firmware is only released as stable for the MK4. However, recently an alpha version for the Prusa Mini has been released. This doesn't quite match the speeds of the MK4 version, but it's a significant increase on the previous Prusa Mini speed. We have not tested this out yet, but if they are able to do this, it will be a particularly impressive achievement to squeeze this additional speed out of an older printer.

In our original review of the Prusa MK4, we said that it was a good printer without the input shaping, and all the things that made it a good printer then still stand. However, it's now almost twice the speed, and that's a great new feature to have.

VERDICT

Everything we loved about the MK4, but now faster.

10/10

LibrePCB v1.0

A new electronics design tool

FREE | **librepcb.org**

By Ben Everard

PCB design is fundamentally a deceptively tricky thing. It's deceptive because the end results aren't usually particularly complex – you just create a few vector images to send off to a factory and, compared to most vector images, they're pretty simple. There aren't that many traces on a PCB compared to, say, brush strokes on a painting.

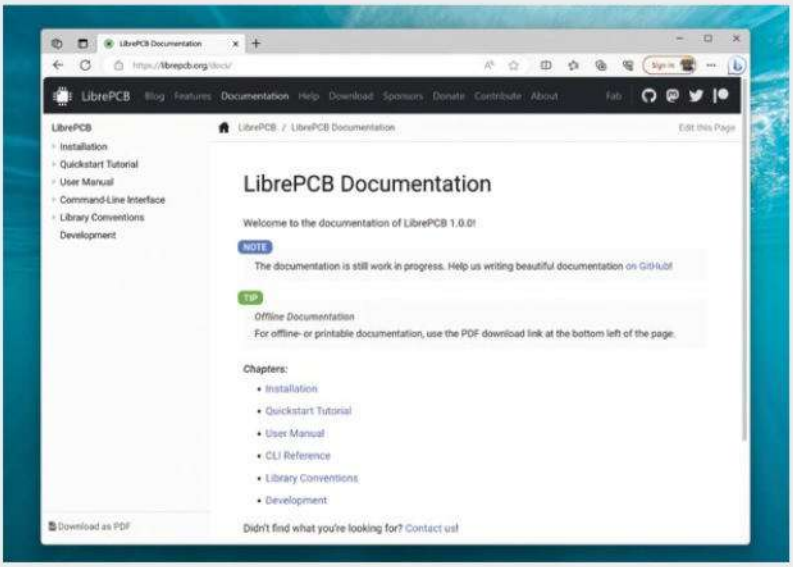
However, the complexity comes not from the amount of detail needed, but the correctness needed. There are thousands of different electrical components, all with slightly different requirements. There are also different limitations and considerations of the various different manufacturing processes to go through, and there are the considerations around the physics of moving electrons through a substance. In other words, there's a lot that can go wrong on a PCB, and it's the job of the design tool to help the designer avoid as many of these problems as possible, while still allowing them the freedom to create the design they want.

Having just reached version 1.0, LibrePCB is much younger than most established players, and this brings both advantages and disadvantages.

LibrePCB feels like it's been designed for flexibility. Footprints are assigned to components after the schematic stage, so you can throw whatever you like into your circuit without having to worry about what part will actually do the job. You can keep several different designs of the PCB synchronised with the schematic, meaning that you can hack about on different designs before deciding which route to go down.

The flexibility extends down to the library level as well, allowing you, for example, to have multiple footprints for the same package. This could be a hand-solderable footprint and a reflow footprint that you can choose between.

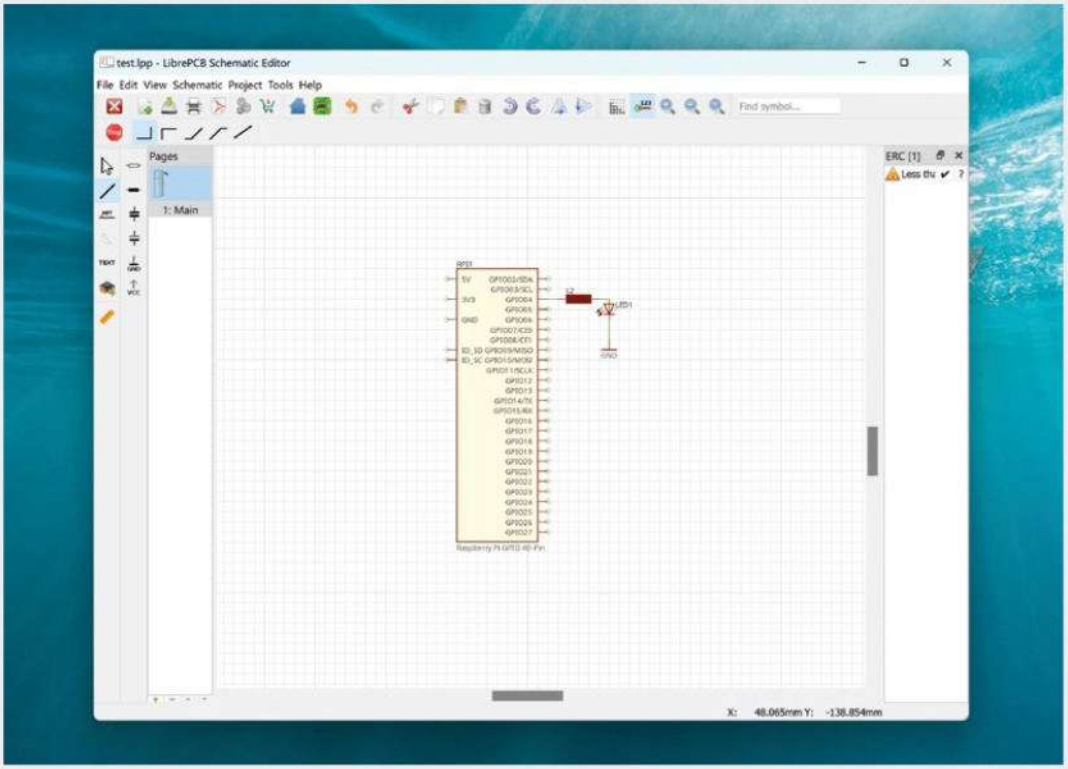
Below
The documentation is currently quite limited, but hopefully it will grow over time



FABRICATION

Perhaps one of the most confusing things about getting a PCB made is Gerber files. This is almost – but not quite – a standard format. It's just different enough that a single Gerber file won't always work between different manufacturers. Or, it might.

LibrePCB includes a fabrication service which allows you to send your projects directly to manufacturers and, in turn, these manufacturers make a donation to LibrePCB. At the moment, this



Left
You too can design a Raspberry Pi HAT with a single LED

only works with AISLER and PCBWay. Depending on your point of view, this could be a convenient way of supporting your design tool while getting your PCBs made, or it could be a commercial intrusion into your software. We're inclined to view it in the first way, and it's by no means the first PCB design tool to be funded by a PCB manufacturer.

There are a couple of downsides to LibrePCB being so new. The first is that it hasn't yet built up as big a library of parts as some of the more established EDA tools. Again, how big a problem this is depends very much on how you use it. Many people prefer to maintain their own libraries anyway as this can reflect the parts they have, and they can be completely confident in the parts there. Trusting a user-contributed library part when designing a PCB is something that's caused us (and many other people) problems in the past.

The second downside is that there is far less documentation about it. There's a 'Quick Start' tutorial and a fairly sparse user guide. You can figure out quite a bit by clicking around, but there are always going to be things you can't find, or figure out how to do.

> **Software without documentation is going to end** up being frustrating sooner or later

Software without documentation is going to end up being frustrating sooner or later. There is a Discourse forum for help, but there's not a lot of activity on it.

We found LibrePCB to be easy to use and generally pretty sensible, but we're quite light hobbyist users. Existing tools, such as KiCad and EAGLE, have plenty of powerful features that are lacking here. How much you miss them will depend entirely on how you use your existing tool.

The LibrePCB developers say that they have created a tool for everyone. We're not quite so sure. We strongly suspect that there are plenty of people out there who like the rigidity of some EDA programs. However, we really like it, and we suspect that it will appeal to others as well.

VERDICT

It's still young, but shows promise.

8/10

sinclair

Commodo

the COMPUTERS THAT MADE BRITAIN

OUT NOW

"The Computers That Made Britain is one of the best things I've read this year. It's an incredible story of eccentrics and oddballs, geniuses and madmen, and one that will have you pining for a future that could have been. It's utterly astonishing!"

- **Stuart Turton**, bestselling author and journalist

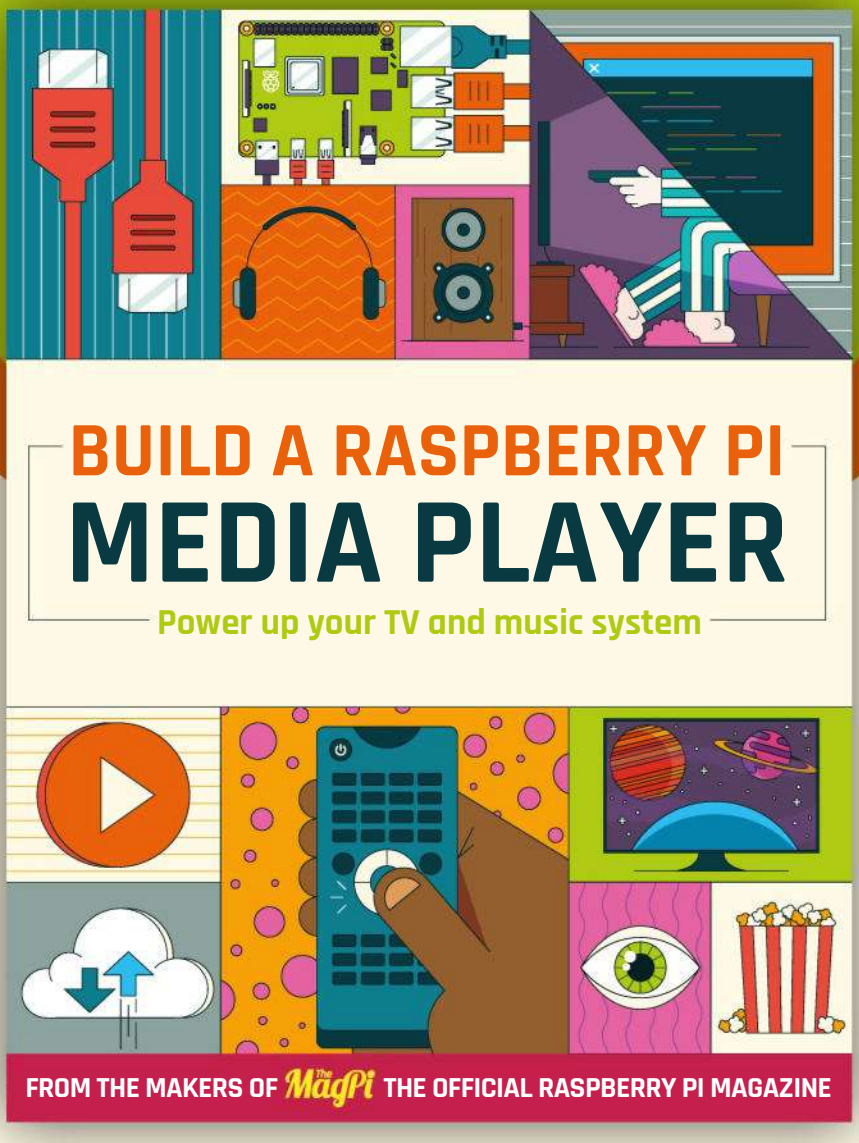




Available on amazon

Buy online: **wfmag.cc/ctmb**

Your FREE guide to making a smart TV



magpi.cc/mediaplayer

Kaluma

JavaScript on Pico

Free | **Kalumajs.org**

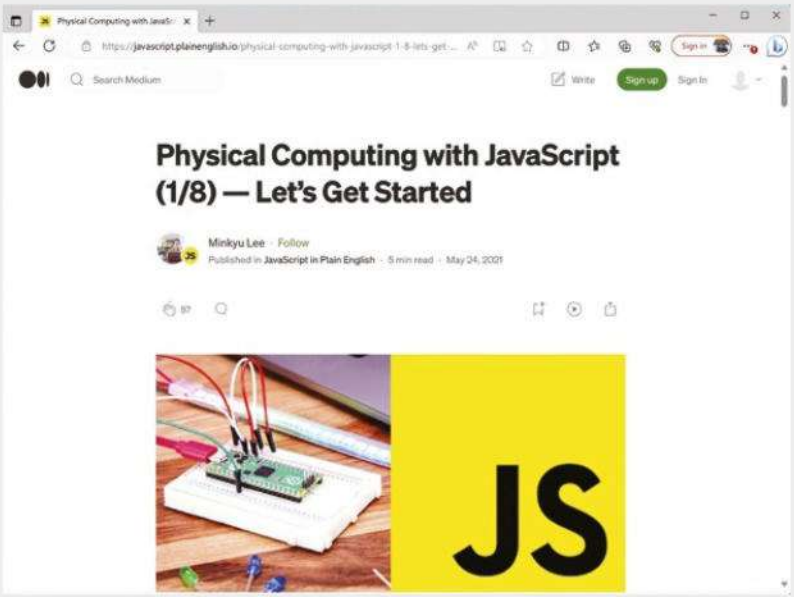
By Ben Everard

Programming languages are strange things. In a very real sense, they're all exactly equivalent – you can write equivalent programs in just about any modern language. However, at the same time, they're entirely different. Learning a programming language – not just the basics, but really taking the time to become fluent in it – affects the way you program, and even the way you think.

Kaluma is a port of JavaScript to RP2040-based devices (currently Pico and Pico W are supported).

It might seem a little strange to port JavaScript to a microcontroller. After all, this is a language that started out as a way to add widgets to webpages, and superficially, that's got little in common with blinking LEDs and reading temperature sensors.

Below
You can find out everything you need to know about getting started at kalumajs.org



JavaScript programs tend to rely heavily on events. Let's take a look at the example of blinking an LED. Using C or MicroPython, you'd typically do this using a loop and a delay, but let's take a look at how it's done in Kaluma:

```
var led = 25;
pinMode(led, OUTPUT);
setInterval(() => {
  digitalToggle(led);
}, 1000);
```

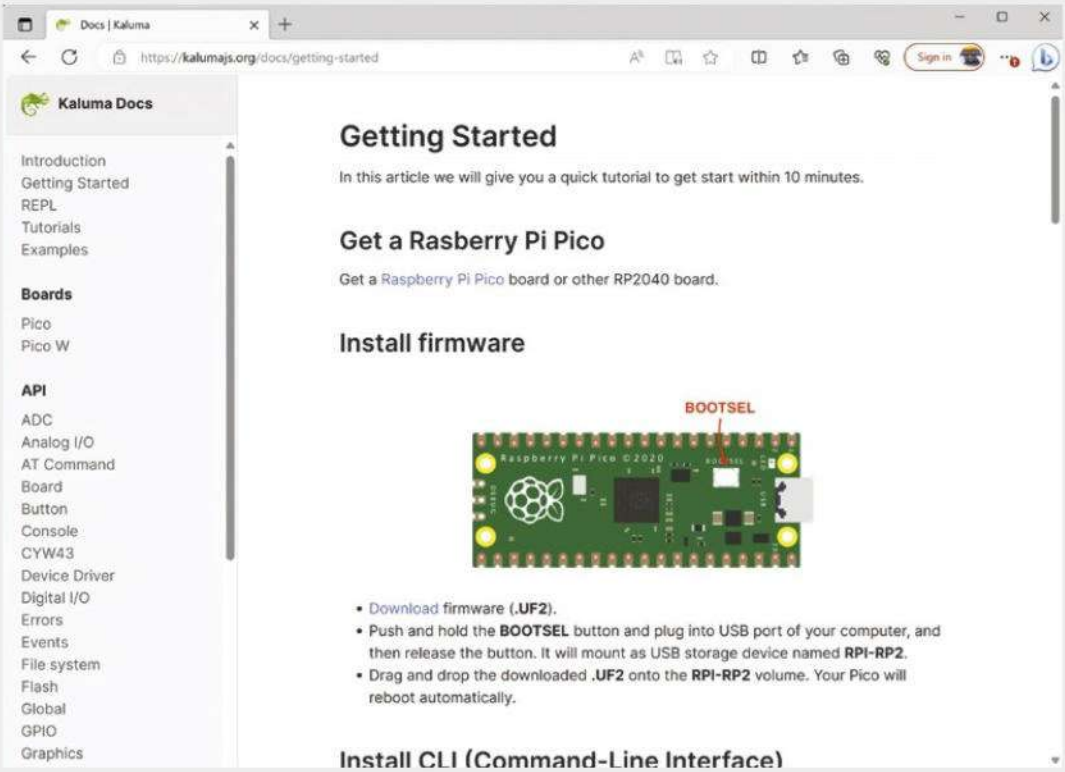
As you can see, the `setInterval` function creates a repeating timer that runs every second and calls another function (in this case, the second function isn't named and just contains one line of code: `digitalToggle(led)`.

You can do this in C, or you can do it this way in MicroPython. But for most C or MicroPython programmers, it wouldn't be the natural way of doing it. Which is the better way of blinking an LED? Well, that's a can of worms that we're not going to open. However, there are two key advantages of having JavaScript on a microcontroller. Firstly, it enables programmers who are most familiar with JavaScript (and other similar) languages to feel at home using microcontrollers. Secondly, it enables people using microcontrollers to gain experience doing things the JavaScript way while still feeling at home. Both of those things are unequivocally a good thing.

EXPLORING KALUMA

This doesn't really say much about Kaluma itself, though, just the idea behind it – so let's dive in and see what it's like.

There are two parts to Kaluma: the firmware that runs on the device (which you can download from **kalumajs.org**) and the software that interacts with



Left The eight-part 'getting started' series will help you start working with hardware

it (which you can install via Node.js). Both installed without problems for us.

With so many things that we look at, documentation is the difference between an OK product and a great product. After all, it doesn't matter how well something could work if you can't work out how to use it. We've been really impressed with Kaluma so far. It's obviously a small product, so don't expect the sheer volume of documentation that you might find with larger ones. However, the API is clearly detailed at **kalumajs.org/docs**, and there's an eight-part series to get you started at **kalumajs.org/docs/tutorials**. This goes through the basics of connecting to common hardware in great detail and explains the technical details behind them.

> **Documentation is the difference between** an OK product and a great product

This series is probably more suited to people who know JavaScript and need some help with the physical computing side than electronic engineers who need help getting to grips with JavaScript.

The main downside of Kaluma is the main downside of every new programming language for a microcontroller – hardware support. By this, we don't mean the hardware on the Pico (or Pico W) boards, but any additional hardware your project needs. At the moment, there's out-of-the-box support for a relatively small amount of hardware. There's not even support for the almost ubiquitous WS2812 LEDs. That isn't to say that you can't use this hardware with Kaluma; you just have to be willing to work at a low level. You can use I2C, SPI, and even RP2's PIO blocks to communicate with most things, but this means working at a low level.

Overall, we found it easy to get started with Kaluma. The software worked and the documentation was easy to follow. It won't be for everyone, but it's a welcome addition to the Raspberry Pi Pico family.

VERDICT

Documentation and tutorials make it easy to get started with this version of JavaScript.

8/10

CROWDFUNDING NOW

InvenTABLE

Cardboard cutting made easy

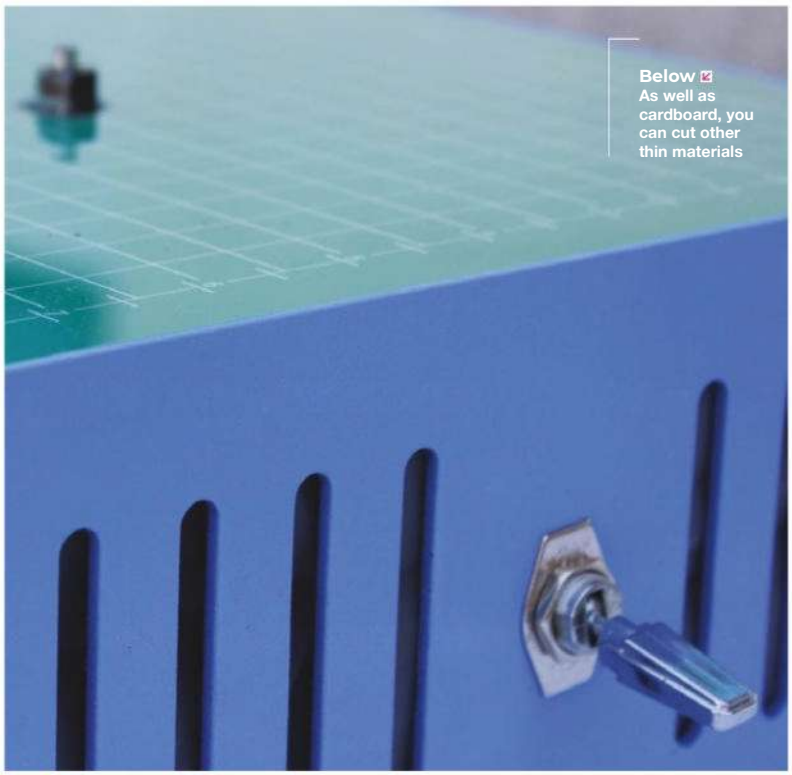
From $189 | hsmag.cc/inventable | Delivery: March 2024

Cardboard is a really great material. It's generally a waste material, so it's basically free, it's strong enough to hold its shape even in large objects, and it's light enough to be easy to work with (even for children). However, it's kind of a pain to cut. Sure, scissors will get through it, but it's awkward to make long cuts without creasing the sheet. You can use a sharp knife, but that's also tricky (and not an option for young makers).

The InvenTABLE is intended as a solution to this. It's a power tool for cutting cardboard in a way that's safe and easy. As far as we can tell, it's similar in style to a sheet metal nibbler, and each oscillation of the cutting head 'nibbles' off a small section without exposing a blade or teeth. This means that it should be very difficult to hurt yourself with it.

At $189, it's not the cheapest way of cutting cardboard. But this should still be within reach of many craft centres and maker spaces (and a lot cheaper than a laser cutter, which is something we've resorted to in the past).

We've not used this, so can't comment on how well it works but, as far as we can see, the principle behind it is straightforward and it looks like it should be possible.



Below As well as cardboard, you can cut other thin materials

BUYER BEWARE

When backing a crowdfunding campaign, you are not purchasing a finished product, but supporting a project working on something new. There is a very real chance that the product will never ship and you'll lose your money. It's a great way to support projects you like and get some cheap hardware in the process, but if you use it purely as a chance to snag cheap stuff, you may find that you get burned.



Above Build your projects out of old packaging

next month

issue #73

ON SALE
23 NOVEMBER

ROBOTS

ALSO

- RASPBERRY PI 5
- 3D PRINTING
- MICROPYTHON
- KICAD
- AND MUCH MORE

DON'T MISS OUT

hsmag.cc/subscribe



Stereovision

Raspberry Pi 5 comes with two camera ports. You can use them for all sorts of things, such as getting images from two different types of camera (like many smartphones do) and capturing data in different directions. However, our favourite use of them so far is taking stereoscopic images. These allow you to capture 3D images. You can process these to produce depth maps or even process them so you can view the images using red/cyan glasses.

HiPi.io

PiKVM

Manage your servers or workstations remotely

A **cost-effective** solution for data-centers, IT departments or remote machines!



PiKVM HAT
for DIY and custom projects



Pre-Assembled version

- Real-time clock with rechargeable super capacitor
- OLED Display
- Bootable virtual CD-ROM & flash drive
- Serial console
- Open-source API & integration
- Open-source software

Available at the main Raspberry Pi resellers

core ELECTRONICS · PiShop.ca · RPishop.cz · Welectron. · elektorstore · PIMORONI

Pi-Shop.ch · ThePiHut · PiShop.us · CloudFree · PiShop.mx

Reseller suggestions and inquiries:
wholesale@hipi.io